

Prise de messages par arbre de capture dans un système transactionnel

THÈSE

présentée à la Faculté des sciences de
l'Université de Genève
pour obtenir le grade de docteur ès sciences, mention informatique

par

Jean-Philippe **BERNEY**

de

l'Abbaye (Vaud)

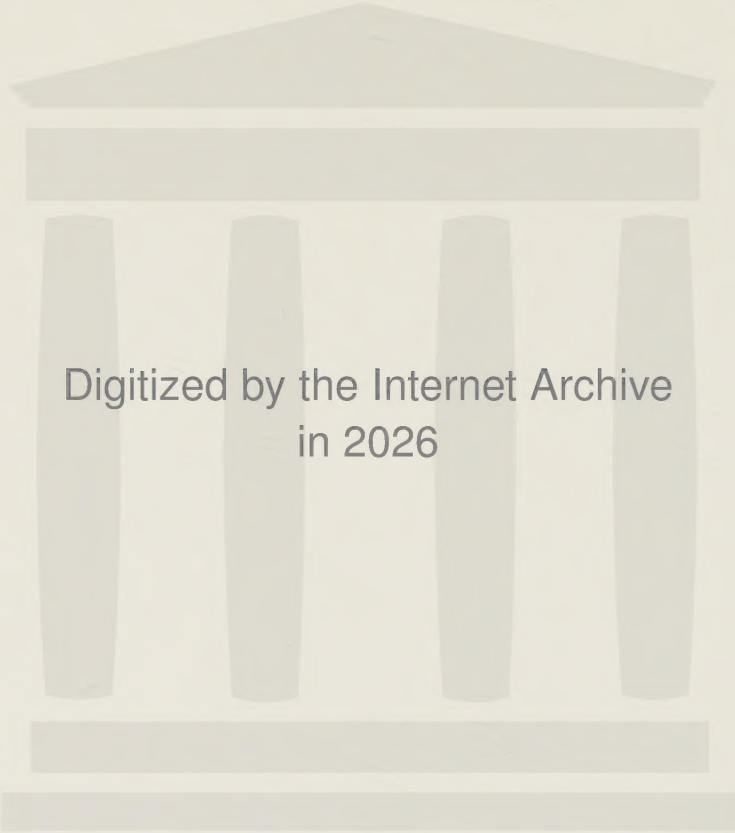
THÈSE No 1844



Juris Druck + Verlag Zürich

1977





Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41545915_0033

Prise de messages par arbre de capture dans un système transactionnel

THÈSE

présentée à la Faculté des sciences de
l'Université de Genève
pour obtenir le grade de docteur ès sciences, mention informatique

par

Jean-Philippe BERNEY

de

l'Abbaye (Vaud)

THÈSE No 1844



La faculté des sciences, sur le préavis de Messieurs
B. LEVRAT, professeur ordinaire, R. BAUD, docteur
ès sciences (faculté de médecine) codirecteurs de
thèse, J. HARMS, professeur ordinaire (Centre univer-
sitaire d'informatique) et J.R. SCHERRER, chargé de
cours (faculté de médecine), autorise l'impression de
la présente thèse, sans exprimer d'opinion sur les
propositions qui y sont énoncées.

Genève, le 21 octobre 1977

Le Doyen :
Bernard LEVRAT

Thèse No 1844



ISBN 3 260 04373 X

REMERCIEMENTS

=====

L'appui bienveillant et la compréhension de plusieurs personnes m'ont été nécessaires pour mener à bien le travail présenté dans cette thèse. Le professeur B. Levrat a accepté de diriger ma thèse. J'ai apprécié la pertinence de ses suggestions et de ses critiques au cours de maintes discussions concernant, en particulier, les aspects théoriques développés au chapitre 2. Monsieur R. Baud, responsable de l'Unité software de la Division d'informatique de l'Hôpital cantonal où ont été réalisés les produits décrits, a participé à toutes ces discussions et a suivi mon travail avec une attention persévérante dont j'ai largement profité. Le docteur J.R. Scherrer, responsable de la Division d'informatique, a accordé d'emblée tout son soutien à l'idée de cette thèse dont, pourtant, on aurait pu craindre qu'elle contribuât à me distraire de l'objet essentiel de mon travail : l'étude et la réalisation pratique des produits supportant l'arbre de capture du projet Diogène. Je suis d'autant plus sensible à la compréhension dont j'ai bénéficié de toute part, que j'avais moins de titre à y prétendre, étant employé, non par l'Hôpital, mais par Control Data (fournisseur des ordinateurs CYBER 173 utilisés dans le SIH Diogène) qu'un contrat lie à l'Hôpital cantonal pour la prestation de services dans le développement de software, sous la conduite et la responsabilité techniques de l'Hôpital.

Dans sa conception générale, l'arbre de capture Diogène doit beaucoup à MEDISCOPE, [CDC 70]*, développé par Control Data pour servir aux travaux de L.L. Weed et J.R. Schultz, [Schultz et al. 71]. Par rapport à ces idées de base, le produit élaboré pour Diogène offre des extensions et des modifications considérables, tenant au haut niveau de confort et de flexibilité recherché, et à l'environnement - logiciel et matériel - totalement différent dans lequel il s'agit d'évoluer. Nombre d'idées en fait d'extension et de confort résultent des discussions que j'ai eues au stade de l'analyse générale, avec les docteurs J.R. Scherrer et A. Assimacopoulos, ainsi que MM. R. Baud et C. Revillard.

La syntaxe ACAP du message transmis aux programmes d'application a été imaginée par MM. R. Baud et B. Gilliard.

M. Ph. Rossier a réalisé le compilateur de pages DESPAC, avec le concours, au départ, de Mlle M. Wintsch et, en fin de travail, de M. J.M. Lugin qui a entrepris de nouveaux développements visant à en améliorer les performances.

MM. B. Gilliard et M. Meier ont écrit certaines portions du code de la chaîne de tâches EXAC permettant l'exploitation d'une librairie de pages.

*voir aux REFERENCES

Tant de suggestions précieuses et de critiques constructives m'ont été faites par les analystes d'application du projet Diogène rédigeant les pages d'arbre de capture et écrivant les chaînes d'application, que je ne saurais énumérer le nom et les apports de chacun.

MM. D. Aebischer, J. Estreicher, A. Mazzario et E. Messmer ont pris la peine de lire les deux premiers chapitres du manuscrit de cette thèse; j'ai fait grand cas de leurs remarques.

Enfin, Mlle D. Lorenzo a dactylographié tout ce texte avec une diligence que je me plais à souligner.

Que toutes les personnes citées à titre personnel ou collectif trouvent ici l'expression de ma très vive gratitude.

Genève, juin 1977

Table des matières

	<u>Page</u>
Introduction	1
1. Présentation de l'arbre de capture Diogène	3
1.1 La prise de messages de transaction en milieu hospitalier	3
1.2 Démonstration sur papier	4
1.3 Arborescence externe d'un arbre de capture	5
1.4 Caractéristiques générales de l'arbre de capture Diogène	6
1.4.1 Diverses formes de messages générés, affichages et pages	
1.4.2 Validation du message de transaction, visa	7
1.4.3 Catégories de chaînes d'application	8
1.4.4 Classification des pages	10
1.4.5 Constitution de librairies de pages	11
1.4.6 Réalisation de la descente dans l'arbre de capture	13
1.4.7 Interface applications / EXAC	15
1.4.8 Correction du message de transaction, descente fictive	16
2. Aspects théoriques	19
2.1 Economie de la prise de messages	
2.1.1 Economie de la prise de messages par arbre de capture en milieu hospitalier selon Le Beux et al.	
2.1.2 Apport éventuel des formulaires affichables	20
2.1.3 Une analyse sémiologique	22
2.1.4 Résumé et conclusion de l'analyse sémiologique	27
2.2 Recherche d'un arbre de capture optimum	28
2.2.1 Formulation du problème d'optimisation	
2.2.2 Reformulations du problème	29
2.2.2.1 Codage optimum des messages de transactions	
2.2.2.2 L'arbre de capture en tant que questionnaire	31
2.2.2.3 Arbre de capture et recherche d'information par une arborescence optimale	34
2.2.3 Possibilités d'optimisation	35
2.2.3.1 Optimisation sans contrainte, algorithme de Huffman	
2.2.3.2 Insuffisances sémantiques de l'optimisation huffmanienne	37
2.2.3.3 Prise en compte de la sémantique	39
2.2.3.4 Optimisation partielle d'une arborescence existante	41
2.2.4 Notations utilisées dans le paragraphe 2.2	43

2.3	Graphe de capture et production du message	46
2.3.1	Sommaire de la syntaxe ACAP	
2.3.2	L'automate de capture	47
2.3.3	Graphes déduits du fonctionnement de l'automate	50
2.3.3.1	Enchaînement de pages et graphe de capture	
2.3.3.2	Arborescence externe	52
2.3.3.3	Description structurelle du message	
2.3.4	Graphe de capture produisant une classe donnée de messages	54
3.	Constitution de librairies de pages	61
3.1	Rôle de la librairie des pages dans le système Diogène	
3.2	Structure des librairies de pages	62
3.3	Format source DESPAC	65
3.3.1	Aperçu de la syntaxe	
3.3.2	Commentaires et aperçu sémantique	67
3.4	Réalisation de DESPAC	72
3.5	Entretien des librairies de pages	
3.6	Réalisation de CHAPAC	73
4.	Réalisation de la descente	74
4.1	Interactions avec l'opérateur	
4.1.1	Nature des messages physiques	
4.1.2	Origine des affichages	75
4.1.3	Affichages spéciaux	
4.1.4	Envoi des affichages EXACi	76
4.2	Interactions avec les autres chaînes	
4.2.1	Une chaîne d'application appelle EXAC	77
4.2.2	EXAC appelle une chaîne d'application	78
4.3	Mécanismes intervenant pendant la descente dans l'arbre	
4.3.1	Allure du CB sous EXAC, structure générale des messages	
4.3.2	Descente normale	84
4.3.3	Mise en route de la descente	86
4.3.4	Débranchement vers des TL extérieures à EXAC	
4.3.5	Descente fictive	87
4.3.6	Enchaînement de transactions	90
4.3.7	Rétrogradation	
4.3.8	Correction ponctuelle, descente automatique	92
4.4	Tâches constitutives d'EXAC	93
4.4.1	Contraintes agissant sur le découpage en tâches	
4.4.2	Rapport avec l'économie de la prise de messages	94
4.4.3	Rôle des diverses tâches d'EXAC. Schéma bloc	97

	<u>Page</u>
4.5 Codification d'EXAC	100
4.5.1 Choix d'un langage	
4.5.2 Effort requis	
5. Bilan et perspectives	102
Références	104
Annexe A : Figures de la démonstration sur papier (paragraphe 1.2)	
Annexe B : Terminologie des graphes	
Annexe C : Répertoire de sigles; glossaire, index	
Echantillon photographique	

INTRODUCTION

=====

Diogène (Division d'Informatique de l'Hôpital de GENEVE) est le nom du projet de Système d'Information Hospitalier (SIH) de l'Hôpital cantonal de Genève. Un tel système devrait permettre de tirer un meilleur parti des ressources de l'Hôpital grâce à la saisie, au traitement et à la diffusion de l'information, en temps réel. Un exposé des motifs est donné dans l'article Scherrer/Baud 73 .

Dévolue à un aspect particulier de ce projet vaste et complexe, cette thèse traite de la prise des messages de transaction temps réel par arbre de capture (sélection d'information sur une suite d'affichages présentés à l'écran d'un terminal interactif), autrement dit de l'interface entre le SIH et ses utilisateurs; encore ne s'agit-il que de l'outil proprement dit, servant à la saisie des messages par arbre de capture, et non du contenu même des affichages intervenant dans tel ou tel domaine particulier des activités médico-hospitalières prises en charge par Diogène.

Limité au regard des applications du SIH Diogène, le sujet ne l'est pas, en revanche, quant au genre de système transactionnel auquel l'interface s'applique. On explique, certes, le succès des arbres de capture dans les SIH par les contraintes propres à l'implantation de tels systèmes, mais rien n'en interdit l'emploi dans d'autres domaines : bancaire, aérien, etc; d'autant moins que les produits décrits ici englobent aussi les facilités usuelles offertes sur des clichs avec champs protégés ou non, présentés sous diverses formes.

Les programmes supportant l'arbre de capture Diogène sont opérationnels et largement testés. Ils sont décrits dans leur état à fin 1976; ils représentaient alors un travail cumulé de sept hommes-années environ, commencé en décembre 1973. Cet effort important, probablement assez rare dans la mise en place de l'interface d'un système transactionnel, s'explique par la volonté des promoteurs du SIH Diogène d'assurer une qualité satisfaisante aux informations saisies en temps réel, et par les circonstances qui ont fait que le développement de l'interface a pu commencer longtemps avant celui des applications proprement dites. Les produits permettant de décrire, d'entretenir et d'exploiter un arbre de capture forment donc un tout autonome, indépendant de toute application spécifique.

L'exploitation de l'arbre de capture (descente dans l'arbre) se fait en temps réel. Il importe de limiter la taille mémoire et de veiller aux performances des programmes sollicités en cours de descente. Ces programmes (ou tâches) sont écrits en assembleur CDC 6000/CYBER. Leur portabilité s'en trouve

naturellement réduite, mais ceci n'enlève rien à la possibilité de reprendre les méthodes utilisées et de les appliquer sur d'autres machines ou systèmes.

L'expression "arbre de capture" utilisée dans le projet Diogène dès son début, est suggérée par le choix répété d'une branche parmi d'autres possibles sur les affichages successifs apparaissant au cours d'une descente. On trouve dans les publications en anglais les expressions "frame selection system", "menu selection", "menu tree", "method of sequential selection"; l'équivalent français de la première de ces expressions est repris dans l'article [Le Beux 75] où il est question de "grilles de sélections".

S'agissant du contenu de cette thèse, le chapitre premier présente les caractéristiques générales des produits supportant l'arbre de capture. L'annexe A complétant ce chapitre, démontre l'emploi de l'arbre de capture et fournit la définition concrète des principaux termes utilisés par la suite.

Le chapitre 2 est un peu plus théorique. La première partie dégage les causes essentielles de l'économie que, du point de vue du donneur d'ordre, représente l'usage de l'arbre de capture par rapport aux moyens conventionnels de prise de messages. Cette économie ne déploie tous ses effets que si l'arbre de capture utilisé est optimum; le second paragraphe est consacré aux méthodes d'optimisation d'arbres de capture. La dernière partie du chapitre part d'une présentation formelle du mécanisme de descente dans l'arbre, pour établir la relation qui existe entre la structure d'un arbre de capture et celle des classes de messages qu'il génère. Ce chapitre 2 fait référence à la théorie des graphes. L'expérience montre que les termes qui s'y rapportent n'ont pas la même signification pour tout le monde; l'annexe B précise les définitions adoptées dans ce contexte.

Les chapitres 3 et 4 donnent un compte rendu relativement détaillé des méthodes et solutions retenues dans la réalisation des produits permettant de créer et d'exploiter l'arbre de capture du SIH Diogène.

Pour éviter au lecteur des recherches fastidieuses, les définitions ou acceptions particulières, de même que les principaux sigles, ont été répertoriés dans l'annexe C.

1. Présentation de l'arbre de capture Diogène

1.1. La prise de messages de transaction en milieu hospitalier

Un système d'information hospitalier (SIH) est un modèle (au sens mathématique) de l'hôpital où il est installé; ceci est naturellement vrai de tout système transactionnel relativement à l'institution qui l'abrite [Nijssen 74].

Comme le système ne peut prendre lui-même ses renseignements, il faut, pour que le modèle évolue suivant la réalité, que les informations voulues lui soient données, en général par le personnel de l'institution, au moyen de terminaux par exemple.

Ces informations sont fractionnées en messages de transactions. La caractéristique principale d'une transaction est de correspondre, dans le système, au déroulement d'une chaîne de programmes d'application qui fait passer la banque de données d'un état cohérent au suivant. Les mécanismes de sauvegarde et de reprise du système sont précisément conçus de telle sorte que, d'un point de vue logique, il n'existe pas d'états intermédiaires de la banque de données où une transaction ne serait que partiellement exécutée.

Dans ces conditions la banque de données représente les états successifs de l'institution réelle alors que les programmes d'application sont une transcription plus ou moins grossière des processus qui y interviennent.

Quoi qu'il en soit, pour la réussite d'un système transactionnel et singulièrement, celle d'un SIH, il est essentiel que la prise des messages de transactions, autrement dit le transit de l'information de la réalité vers le modèle, soit aisé et sûr, donc efficace ou encore économique au sens le plus large du terme.

A plusieurs reprises, ces dernières années, il a été fait état des avantages, en milieu hospitalier tout au moins, de la prise de messages de transaction par arbre de capture, et de réalisations dans ce domaine : [Schultz et al. 71] (travaux de L.L. Weed et J.R. Schultz au Medical Center Hospital de Burlington, Vermont), [Bowden et al. 73], [Le Beux et al. 74], et auparavant, [Barnett et al. 68].

Que ces vues se soient imposées essentiellement en milieu hospitalier tient non seulement à la mode mais aux contraintes particulières de cet environnement :

- entrée des messages par du personnel sans formation ni vocation informatiques;
- terminologie médicale particulièrement riche et d'orthographe difficile;

- nécessité de messages exempts d'erreurs vu la portée médicale des transactions qu'ils induisent.

Le chapitre 2 montrera que la méthode de l'arbre de capture (AC) permet de surmonter la plupart des difficultés résultant de ces contraintes.

La suite de ce premier chapitre présente l'AC Diogène de manière concrète par une démonstration sur papier, puis en décrit les caractéristiques générales tout en précisant la terminologie qui s'y rapporte.

1.2. Démonstration sur papier

Pour l'utilisateur au terminal, le fonctionnement de l'AC Diogène est d'une extrême simplicité : étant donné un affichage à l'écran du terminal, il suffit à l'opérateur de choisir un élément dans une énumération d'objets bien distincts; ce choix provoque l'envoi de l'affichage suivant où l'opérateur choisit un élément de la nouvelle énumération envoyée et ainsi de suite. Chaque énumération est pourvue de commentaires facilitant la décision. Le geste requis pour marquer un choix est réduit au minimum : selon le type de terminal il s'agit de toucher sur l'écran l'élément voulu de l'énumération ou d'envoyer par le clavier du terminal le caractère alphanumérique étiquetant l'objet dans l'énumération.

Les figures de l'annexe A reproduisent deux descentes d'AC; la suite des cadres est rigoureusement celle des affichages apparus au terminal. Elle est reprise d'un fichier généré automatiquement en parallèle avec toute descente en période de test. Il est à peine besoin de souligner l'intérêt d'un tel fichier lors des discussions dans les groupes d'analyse, ou dans la recherche des causes d'erreurs et la formation des utilisateurs du système transactionnel.

Chaque cadre représente donc un affichage, de 24 lignes à 80 colonnes dans la version utilisée pour la démonstration; les textes des choix sont disposés sur deux colonnes et étiquetés de "A" à "X", et n'apparaissent qu'aux lignes de numéro pair. Une marque >>> à gauche, ou <<< à droite d'un cadre indique le choix fait sur cet affichage en cours de démonstration. Parfois il a été fait usage d'une "touche fonction" signalée à droite au bas du cadre.

Les affichages contiennent en général des lignes de commentaires et font apparaître le message de transaction (ou tout au moins l'une des formes de ce message) en haut de l'écran.

Pour étoffer la démonstration l'opérateur s'est plu à compliquer la descente par des corrections et repentirs; normalement une descente se passe de manière plus simple.

1.3. Arborescence externe d'un arbre de capture

La démonstration précédente révèle que, vu de l'extérieur, par exemple d'une console de visualisation, l'AC se présente comme une sorte de questionnaire à choix multiples dont les questions sont enchaînées automatiquement, compte tenu des réponses aux questions antérieures. Il est provisoirement fait abstraction des pages de type-C (clichés ou affichages où l'opérateur peut entrer des données dans des champs non protégés); quel que soit le confort offert dans leur usage, elles demeurent étrangères au principe général de l'AC. Sur les pages de type-S, (affichages sur lesquels l'opérateur fait une sélection ou choix) qui forment l'arbre de capture proprement dit, l'envoi d'une réponse par l'opérateur provoque l'apparition de l'affichage suivant (question suivante) à moins que, selon des critères précisés plus loin, le message de transaction soit complet; en première approximation, ce message, du moins l'une de ses formes, résulte de la concaténation des textes de réponses sélectionnés sur les affichages successifs. Suivant cette définition l'AC est représentable par une arborescence que nous dirons externe, dont les sommets correspondent aux affichages susceptibles d'être présentés, et les arcs sortant d'un sommet donné, aux choix offerts sur l'affichage associé au sommet considéré; un seul de ces arcs est emprunté au cours d'une descente, représentée par le chemin joignant la racine de l'arborescence à un sommet terminal donné. L'annexe B précise la terminologie adoptée ici en théorie des graphes.

Un AC est représenté convenablement par son arborescence externe lorsque l'on ne se préoccupe pas du mode d'enchaînement des affichages successivement présentés. Ce sera le cas au paragraphe 2.2 quand il s'agira de rechercher, étant donnés des messages de transaction, des arborescences optimales susceptibles de les produire.

Si, par contre, on veut traduire ne serait-ce que les deux faits suivants, fondamentaux dans l'AC Diogène, à savoir

- qu'un même affichage peut servir de successeur à des pages distinctes,
- qu'un affichage n'est pas forcément désigné comme successeur par son prédécesseur immédiat,

il faut recourir, pour représenter l'AC de façon adéquate, à des graphes de structure moins simple que celle des arborescences. C'est dire que l'expression "arbre de capture" doit être prise comme un tout; elle ne signifie pas qu'il soit représenté convenablement par un "arbre" soit au sens de la définition de ce terme rappelée dans l'annexe B, soit encore au sens que lui prêtent certains ouvrages, par exemple [Knuth 68], qui est celui d'une arborescence selon la définition adoptée ici. Le problème de la représentation de l'AC est traité au paragraphe 2.3.

1.4. Caractéristiques générales de l'arbre de capture Diogène

Ce paragraphe définit de manière plus systématique les termes rencontrés incidemment dans la démonstration sur papier et passe en revue les caractéristiques principales de l'AC Diogène.

1.4.1 Diverses formes de messages générés, affichages et pages

Le message généré, en première approximation par mise bout à bout des informations successivement choisies, constitue la face visible, "en clair", du message de transaction. On le désignera par MAC (Message - Arbre de Capture). C'est lui que voit se former l'opérateur du terminal ou le donneur d'ordre.

De même les affichages successifs représentent la face visible de ce qui constitue l'AC. Il est évident en effet, qu'outre ces informations visibles à l'opérateur, l'AC doit comporter au moins les informations de structure permettant de décider, étant donné une sélection faite sur un affichage, quel est le prochain affichage à présenter. On désignera, en général, par page la combinaison d'un affichage (informations visibles) et des informations de structure (plus généralement, de toutes les informations invisibles) qui lui sont associées. Une page constitue un article sur un fichier de la banque de données : la librairie des pages de l'AC. Il est important que toutes les informations (affichage et informations de contrôle) pour une page donnée soient physiquement groupées de manière qu'un seul accès à la banque de données amène tout ce qui est nécessaire pour faire progresser d'un pas la descente dans l'AC.

Alors que le MAC constitue un message en clair pour l'opérateur, son utilisation par les chaînes de tâches chargées de son traitement en tant que transaction est malaisée. Un message codé est donc produit parallèlement au MAC. Désigné par MBG (Message-Back Ground), il résulte, en gros, de la concaténation structurée d'informations invisibles, dites informations back ground (info BG).

Le MAC et le MBG reflètent le chemin suivi au cours de la descente dans l'AC; cependant, même s'ils le font de manière univoque (au prix de certaines contraintes sur la conception de l'AC), la détermination du chemin suivi, fondée sur l'analyse de ces messages, n'est ni triviale ni économique. Or il est nécessaire de pouvoir reconstituer ce chemin de manière simple et rapide pour permettre la correction partielle du message ou la rétrogradation dans l'AC. On conserve donc une trace explicite du chemin suivi, définie principalement par la suite des identificateurs des pages dont l'affichage a été présenté et des numéros des sélections qui y ont été faites. Ce message est désigné par MMS (Message - MynosotIS).

Au "terme" (compte tenu de toutes les nuances qu'il convient de donner à ce terme) de la descente dans l'AC, MAC et MBG sont délivrés à la chaîne de tâches d'application traitant la transaction proprement dite; le nom de cette chaîne de tâches doit être spécifié par une, au moins, des sélections faites au cours de la descente dans l'AC.

1.4.2 Validation du message de transaction, visa

Tout message généré par une descente dans l'AC est, si cet arbre est conçu de manière correcte, valide du fait même que le message a pu être produit de cette façon. Ce point d'une importance capitale, sera développé au chapitre 2. Cette validation n'est toutefois que partielle dans la mesure où, pendant la descente proprement dite, il est fait, en général, référence aux pages de l'AC dont la majorité est fixe, plutôt qu'aux informations variables de la banque de données, celles concernant un patient déterminé, par exemple.

Cette validation partielle offre les caractères :

- a) d'une validation syntaxique : étant donné une fonction, la descente dans l'AC garantit que le message résultant contient la suite de tous les renseignements ou paramètres (et de ceux-là seulement) nécessaires et suffisants pour définir cette fonction. Risquons ici une analogie grammaticale :

message de transaction	↔	proposition
fonction (p.ex. prescription de médicament)	↔	verbe
paramètres applicables à cette fonction (p.ex. dose, début de traitement, etc.)	↔	compléments d'objet et circonstanciels permis ou impliqués par l'usage de ce verbe

- b) d'une validation sémantique partielle : par une conception judicieuse de l'AC on peut faire en sorte que, dans une prescription de médicament par exemple, il ne soit possible de prescrire que les doses effectivement applicables au médicament considéré; de manière plus générale, par descente dans l'AC, on ne génère que des messages plausibles eu égard à l'état actuel de la science et étant admis que le patient auquel il s'applique est en quelque sorte "standard".

Par contre, en tant que tel, l'AC ne permet pas de tenir compte d'informations étrangères à la librairie des pages; les validations portant sur de telles informations sont le fait de la chaîne d'application particulière appelée au terme de la descente, qui s'applique à la fonction en cause et exécutera la transaction proprement dite. Etant donné le caractère essentiellement sémantique de cette validation finale, la chaîne ainsi appelée est dite chaîne de validation sémantique.

Dans le cas d'une prescription de médicament, la validation sémantique consiste en particulier à vérifier que le médicament prescrit n'est pas contre-indiqué eu égard aux médicaments que le patient a déjà reçus ce qui implique des accès à d'autres fichiers que la librairie des pages.

Après la validation sémantique, l'application responsable du traitement de la transaction rappelle la chaîne permettant la descente dans l'AC, pour affichage du message ainsi validé, en vue d'obtenir le visa du donneur d'ordre (autrement dit la bénédiction ou OK de l'infirmière). En règle générale ce message ne devrait pas différer du MAC généré par la descente dans l'AC avant validation sémantique. Toutefois, la chaîne d'application pourrait apporter au MAC quelques retouches et, de toute façon, pour marquer le rôle que joue ce message à ce stade, on le désigne par MEV (Message - Visa).

Notons que jusqu'à ce point, la banque de données n'a pas subi d'altération physique. Les mécanismes de sauvegarde et reprise du système sont conçus de telle façon que les écritures ne deviennent effectives que lorsque l'opérateur choisit "OK" lors de la demande de visa. Tout autre choix, en cette circonstance, annule les intentions de mises à jour émises antérieurement.

1.4.3 Catégories de chaînes d'application

Le schéma de déroulement d'une transaction que nous venons d'indiquer, à savoir

1. descente dans l'AC, avec validation partielle du message par la descente elle-même,
2. validation sémantique finale par la chaîne de validation sémantique,
3. visa du donneur d'ordre,
4. mise à jour physique de la banque de données,

est apparu d'emblée comme trop rigide. Outre la chaîne de validation sémantique et d'exécution, des chaînes d'application doivent pouvoir intervenir, en cours de descente dans l'AC, entre deux affichages successifs. L'analyse

détaillée et le développement des premières applications de Diogène ont confirmé cette nécessité. On distingue à cet égard

- a. les chaînes de remplissage-S ou -C : elles permettent de présenter sous forme de pages de type-S ou -C respectivement (paragraphes 1.2 et 1.4.4), des informations variables, prises en général de la banque de données. Les pages sujettes à un tel remplissage avant affichage sont connues sous le nom de squelettes;
- b. les chaînes de validation MFL (Message Format Libre) servant à la validation, d'ordinaire purement formelle, des informations introduites par l'opérateur sur une page type-C; naturellement ces chaînes peuvent non seulement valider, mais aussi employer le MFL : le phonétiser, par exemple, et l'utiliser comme clef d'accès à un fichier de noms de médicaments, etc.; en général la chaîne peut demander la correction immédiate des données manifestement erronées que l'opérateur peut avoir introduites;
- c. les chaînes auxiliaires appelées après traitement d'une sélection sur une page type-S permettant, notamment, la validation partielle ou la réorganisation du message de transaction en cours de formation.

Il est évident que le déroulement de telles chaînes en cours de descente nuit à la qualité du temps de réponse (laps de temps entre choix fait par l'opérateur et apparition sur l'écran de l'affichage suivant). Il faut donc trouver un compromis entre l'avantage qu'elles offrent et la surcharge du système qu'elles peuvent causer.

Dans le projet Diogène, les chaînes d'application sont matérialisées par des ensembles de "tâches" ou modules exécutés à la chaîne selon une séquence dépendant de la donnée que constitue, par exemple, le message de transaction, ou le MFL validé, ou encore les informations trouvées sur la banque de données lors du remplissage d'une page squelette. Pratiquement, une tâche a la forme d'un "overlay"; les tâches d'une même chaîne communiquent entre elles par une zone de mémoire commune, le bloc de communication. Fréquemment le sigle TL (pour Task-List) désigne une chaîne de tâches.

1.4.4 Classification des pages

a. Type-S ou -C

Toute page de l'AC relève de l'un ou l'autre de ces types. Les pages de type-S ou pages de sélections constituent l'arbre de capture proprement dit. Les pages de type-C ou clichés permettent d'introduire des informations au clavier du terminal lorsqu'il est malcommode de prendre ces informations par sélections successives sur des pages de type-S. Les pages type-C ont pris une importance croissante avec l'avancement du projet, surtout lorsqu'on s'est avisé de faire de l'AC l'interface exclusive entre applications et donneur d'ordre. Ces pages offrent entre autres facilités :

- une manière simple de spécifier sur l'écran des champs des divers types offerts par le terminal utilisé : protégés ou non, écriture blanc sur noir ou l'inverse, champs clignotants, atténués, etc.;
- la possibilité de préremplir certains champs, par exemple non protégés, par une chaîne de remplissage-C, avant l'envoi de l'affichage, avec des informations variables que l'opérateur peut modifier ou non. Ceci est utile s'il s'agit de compléter ou modifier le remplissage d'un formulaire administratif;
- la possibilité pour une chaîne de validation MFL de provoquer le renvoi de l'affichage prérempli avec ce que l'opérateur vient d'y introduire, en faisant par exemple clignoter les champs suspects et en exhibant un diagnostic explicatif (voir prise de la date dans la démonstration 1.2).

b. Pages dynamiques et squelettes

Une page squelette, en général de type-S, sert, par définition, à présenter des informations variables prises normalement sur la banque de données par une chaîne de remplissage. Lorsque la fréquence de présentation de ces informations variables est nettement supérieure à celle de leur mise à jour, il peut être utile de les garder en permanence sous forme d'une ou plusieurs pages de l'arbre de capture. Ces pages type-S susceptibles d'être modifiées, en cours d'exploitation, par des transactions, sont dites dynamiques. En marge du logiciel supportant l'AC Diogène, un ensemble de routines permet l'analyse et la mise à jour des pages dynamiques par les chaînes d'application. Les pages dynamiques et squelettes posent quelques problèmes quant à la reproductibilité d'une descente dans l'AC (reproductibilité qui doit être garantie pour permettre la correction de message par exemple) et à la cohérence du message de transaction avec la banque de données au moment de la validation sémantique et de l'exécution de la transaction (voir paragraphes 1.4.8 et 4.3).

c. Pages clef

Il est utile de pouvoir produire successivement plusieurs messages de transaction touchant peut-être le même patient et relevant de la même application : plusieurs radiographies ou plusieurs prescriptions de médicaments, par exemple, étant entendu qu'il s'agit bien de transactions distinctes, donc de séquences de mises à jour de la banque de données tout à fait indépendantes les unes des autres. Les pages clef facilitent la production de telles variantes de messages en permettant au donneur d'ordre de réutiliser la partie commune de ces messages (leur début); en cours de descente, l'AC garde, en effet, la trace des pages clef rencontrées, et, après exécution d'une transaction, se replaçant sur la dernière page clef trouvée, l'envoie au terminal. L'opérateur peut soit enchaîner sur cette page, soit en sortir ce qui provoque l'envoi de la page clef précédente et ainsi de suite.

En écrivant les pages d'un AC, il conviendra donc de déclarer comme clefs les pages type-S intervenant aux embranchements les plus importants de la descente.

1.4.5 Constitution de librairies de pages

Les pages de l'AC Diogène forment un fichier : la librairie des pages. Ce fichier doit appartenir à la banque de données car il subit des mises à jour pendant l'exploitation (pages dynamiques); il doit donc bénéficier des mécanismes de sauvegarde des fichiers de la banque de données.

Le chapitre 3 décrit les facilités offertes dans le projet Diogène pour la création et l'entretien des librairies de pages dont la réunion forme la librairie complète servant en exploitation. Ce paragraphe introductif souligne les aspects les plus significatifs de la question.

- a. La description des pages (affichage et informations de contrôle) est présentée dans un format source élaboré qui le rapproche d'un langage. Ce format source DESPAC (DEscription des Pages de l'AC) est soumis au "compilateur" DESPAC qui teste la syntaxe de la description source et produit un fichier de pages compilées c'est-à-dire codées et compactes.
- b. Toutes les pages d'une librairie donnée, autrement dit, d'un AC donné n'ont nul besoin d'être compilées en une fois. Un utilitaire, CHAPAC (CHargeur des Pages de l'AC), se charge de la création d'une librairie formée sur un noyau de pages initial, de l'adjonction à une librairie existante de pages nouvellement compilées, du remplacement de certaines pages par leur dernière version, de la fusion de deux librairies préexistantes, de la restructuration

d'une librairie; enfin, ce même CHAPAC peut faire l'analyse d'une librairie pour en faciliter la mise au point : références croisées des désignations de pages, répertoire des appels de chaînes d'application des diverses catégories etc.

- c. Un problème important est celui de la désignation des pages. Chaque sélection d'une page type-S peut désigner, en général, plusieurs pages successeurs, c'est-à-dire devant être affichées tôt ou tard à l'écran suivant une règle donnée au prochain paragraphe. Quatre types de contraintes agissent sur le mode de désignation des pages :
 1. La désignation en source DESPAC d'une page, soit dans cette page elle-même, soit dans les références qui y sont faites par d'autres pages qui l'ont pour successeur, doit être aussi parlante que possible pour celui qui rédige la description des pages (d'autant plus qu'il s'agit en fait souvent de plusieurs personnes), un peu comme un nom de variable dans un langage de programmation;
 2. Les pages d'une librairie doivent pouvoir être compilées indépendamment les unes des autres et placées dans une librairie existante; donc pour interpréter le nom d'une page il ne doit pas être nécessaire que cette page figure dans le lot de pages en cours de compilation;
 3. Pendant la descente dans l'AC il faut que de la désignation d'une page sous forme compilée, on puisse déduire son emplacement en librairie sans recours à aucun fichier index (question d'efficacité) et sans table volumineuse en mémoire centrale;
 4. Une expression du genre : "désignation source de page" + "nombre", au niveau source DESPAC, doit conserver le même sens en cours de descente. En effet il est très utile de pouvoir spécifier une page successeur à une grandeur additive près, provisoirement inconnue; supposons par exemple qu'on sache au stade actuel de la descente qu'il faudra présenter tôt ou tard une page de commentaire sur tel type d'analyse dont la transaction est précisément une requête, type d'analyse encore indéterminé; la désignation de la page commentaire peut faire intervenir une variable d'indexation, appelée registre software, dont le contenu sera introduit, en fonction du type d'analyse, dans le traitement d'une sélection faite plus bas dans la descente et utilisé au moment de l'affichage de la page commentaire. Toutefois, pour que ce mécanisme joue, il faut que la séquence des pages commentaires d'analyse définie au niveau source, soit retrouvée lors de la descente dans l'arbre.

La structure des librairies de pages et le mode de désignation des pages dans l'AC Diogène sont décrits au chapitre 3.

1.4.6 Réalisation de la descente dans l'arbre de capture

L'exploitation d'une librairie de pages donnée, autrement dit, la descente dans l'AC, est réalisée par une chaîne de tâches connue sous le nom d'EXAC (EXploitation de l'AC). Cette chaîne ne jouit pas de propriétés particulières par rapport aux chaînes d'application définies au paragraphe 1.4.3, si ce n'est que, puisque l'AC sert d'interface entre le donneur d'ordre et les applications, la chaîne EXAC est la première appelée lorsqu'une activité se manifeste à un terminal interactif. En général, selon des modalités dont le détail importe peu, entre deux périodes d'activité (ou sessions) au terminal, celui-ci présente un "affichage de repos" dont le contenu dépend par exemple du genre de transactions demandées le plus souvent de ce terminal. Lorsqu'il reçoit le premier message physique d'une session, EXAC détermine quel est l'affichage de repos du terminal émetteur et peut, en général, interpréter ce message comme une sélection faite sur l'affichage en question.

Ceci amorce la descente dans l'AC, descente qui progresse pas à pas sous le contrôle d'EXAC par l'interprétation de la sélection que vient de faire l'opérateur sur le dernier affichage de type-S envoyé à l'écran. On peut négliger ici le traitement du message entré sur une page type-C. Très schématiquement, interpréter une sélection revient à :

- compléter les messages MAC et MBG par la contribution éventuelle à ces messages, requise par la sélection;
- accumuler sur une pile auxiliaire (file d'attente gérée selon la règle premier entré, dernier servi) l'ensemble des désignations de pages successeurs figurant dans les informations de contrôle de la sélection;
- prélever le sommet de la pile auxiliaire, lire la page qu'il désigne, envoyer l'affichage au terminal;
- attendre la réponse de l'opérateur, c'est-à-dire la sélection faite sur l'écran.

De façon simplifiée on peut dire que la descente se termine lorsque la pile auxiliaire est vide; EXAC appelle alors la chaîne de validation sémantique spécifiée antérieurement, par une sélection. Nous rejoignons ainsi le schéma de transaction du paragraphe 1.4.3. En cours de descente, les sélections sur les pages type-S ou les informations de contrôle dans les pages type-C, peuvent demander un débranchement temporaire vers des chaînes d'application auxiliaires ou de validation

MFL; il se peut aussi (cas des pages squelettes) qu'immédiatement après lecture de la page en librairie, une chaîne de remplissage soit sollicitée pour garnir l'affichage avant son envoi au terminal. Dans tous les cas, ces chaînes concluent leur exécution en appelant EXAC.

Le grand avantage de la pile auxiliaire de désignations des pages est de permettre à certaines pages de ne désigner aucun successeur explicitement et, partant, d'être utilisables à divers endroits dans l'arbre de capture afin d'éviter la prolifération de pages quasi identiques, ce qui causerait des problèmes d'entretien et du gaspillage de mémoire de masse. Le caractère portable d'une page se généralise sans peine à un ensemble de pages : s'il ne désigne pas de successeur en dehors de lui-même, il forme une sorte de sous-arbre appelable n'importe où dans l'arbre de capture global; cette affirmation doit être nuancée si l'on tient compte d'effets globaux que peut avoir sur le contexte un sous-arbre; il peut par exemple altérer le contenu d'un registre software (paragraphe 1.4.5) dont le contenu pourrait avoir été initialisé avant appel du sous-arbre en vue d'un usage postérieur à la sortie du sous-arbre. Ceci n'enlève rien à l'intérêt de la pile auxiliaire mais donne une idée quant aux précautions à prendre dans l'emploi des registres software.

Au sujet de la portabilité des pages, à titre individuel ou en tant que sous-arbre, on relèvera que dans l'AC Diogène le souci d'éviter la multiplication des variantes de pages s'est traduit par des développements assez importants; mentionnons :

- l'existence des registres software dont il vient d'être question; l'idée de ces registres est reprise de MEDISCOPE [CDC 70];
- l'existence de masques applicables sur les pages type-S. Un masque peut être explicite (spécifié dans la sélection même qui désigne comme successeur la page masquée) ou contenu dans un registre software, ce qui permet de modifier dynamiquement le masque, indépendamment de la désignation comme successeur de la page soumise au masque;
- la possibilité d'appeler une page avec répétition sans que cette page se désigne elle-même comme successeur;
- la possibilité de spécifier, dans l'appel d'une page comme successeur, un commentaire devant figurer dans la page appelée; on peut ainsi "personnaliser" une page universelle de prise de la date, par exemple, en prise de la date d'examen, prise de la date de sortie, ce qui assure un meilleur guidage de l'opérateur, donc une qualité supérieure de la prise de message.

On perçoit sur cet exemple quelques règles données ici sans prétention à la rigueur :

- à chaque contribution au MAC correspond, dans le MBG une contribution de la forme "&étiquette=valeur" entre virgules;
- l'étiquette caractérise la nature de la valeur qui lui est associée;
- la valeur peut être simple ou complexe, c'est-à-dire, elle-même composée d'une suite de formes "&étiquette=valeur" mises globalement entre parenthèses.

Cette dernière règle s'applique récursivement et reflète l'existence de sous-arbres de capture gigognes.

Le MBG constitue la partie la plus significative des informations passées par EXAC dans le CB aux applications. En sens inverse, EXAC reçoit de la chaîne d'application qui l'appelle, également dans le CB

- un code indiquant la circonstance dans laquelle EXAC est activé : mise en route d'une descente, retour d'une chaîne de remplissage -S ou -C, retour d'une chaîne de validation MFL demandant le renvoi du cliché pour correction des données entrées par l'opérateur, ou encore envoi de la demande de visa après validation sémantique, etc.;
- des informations complémentaires, par exemple, s'il s'agit du retour d'une chaîne de remplissage-S, ces informations sont celles qu'EXAC doit inclure à la page squelette avant de l'afficher et d'interpréter le choix que fera le donneur d'ordre.

Ces points seront repris plus en détail dans les chapitres suivants.

1.4.8 Correction du message de transaction, descente fictive

Dans la règle, l'opérateur peut à tout moment de la descente, activer des touches fonctions permettant

- de rétrograder (touche =), c'est-à-dire faire défiler les affichages présentés antérieurement dans la descente, puis reprendre la marche en avant du point atteint en reculant;
- d'obtenir l'affichage du MAC tronçonné (touche *) selon les contributions apportées au MAC par les pages présentées successivement, et sur cet affichage spécial, de sélectionner la portion de message à corriger; EXAC envoie alors précisément la page ayant apporté la contribution visée au MAC; la descente peut reprendre de ce point; accessoirement, si le chemin corrigé rejoint le chemin suivi initialement

dans l'AC, l'opérateur peut provoquer (par la même touche *) une descente automatique jusqu'au point atteint avant qu'on ait décidé de corriger.

Ces deux modes de correction font usage d'un processus essentiel (et probablement original) de l'arbre de capture Diogène : la descente fictive, ainsi nommée parce qu'elle se fait sans intervention (normalement tout au moins) de la part du donneur d'ordre et, sous réserve d'un accroissement possible et momentané du temps de réponse, sans même qu'il s'en rende compte. Elle reproduit la descente réelle, de la racine de l'AC au point A d'où l'on veut reprendre la descente réelle.

De façon générale, le but recherché en effectuant une telle descente fictive est un repositionnement dans l'AC en un point A où l'on est passé antérieurement dans une descente réelle, avec rétablissement exact des conditions dans lesquelles on se trouvait en ce point A lors de la descente réelle. Les conditions à reconstituer sont par exemple :

- la pile (et la pile préfixe auxiliaire) y compris les compteurs de parenthèses MBG ouvertes aux divers niveaux;
- l'état des registres software.

Naturellement on peut imaginer d'autres méthodes que la descente fictive pour obtenir le même effet et qui éviteraient, peut-être, la relecture des pages que nécessite la descente fictive. Toutefois, la descente fictive offre deux avantages décisifs :

- elle permet de reconstituer les conditions en A en ne gardant dans le CB qu'une trace minimum (1 entrée MMS de 1 mot par pas dans la descente réelle);
- les routines de traitement utilisées en descente fictive sont pour l'essentiel celles qui servent dans une descente réelle, ce qui simplifie la conception et l'entretien des programmes.

La correction du message de transaction n'est pas seule à faire usage de la descente fictive; l'enchaînement de transactions par le mécanisme des pages clef (paragraphe 1.4.4) demande une descente fictive lorsque l'opérateur fait sur une telle page un choix différent de celui qui provoque l'envoi de la page clef du niveau supérieur.

La réalisation de la descente fictive soulève deux difficultés :

- il faut éviter de réexécuter certaines chaînes auxiliaires ou de validation MFL dont l'exécution proprement dite ou la mise en place des conditions initiales, demanderaient

l'intervention de l'opérateur; si toutefois une telle chaîne doit être réexécutée pour assurer la reconstitution de l'état au point visé par la descente fictive, EXAC l'interrompt et demande le secours de l'opérateur. En principe l'AC peut être conçu de façon que les chaînes appelées en cours de descente soient réexécutées sans problème ou court-circuitées, ce qui évite toute interruption intempestive;

- la descente fictive réutilise les pages présentées en descente réelle; comme ces pages peuvent être dynamiques (paragraphe 1.4.4) et qu'il vaut mieux éviter de poser des verrous sur toutes les pages utilisées au cours de la descente (compte tenu de l'usage simultané de l'AC, et de la banque de données en général, par plusieurs terminaux), on peut craindre qu'une page vue dans la descente réelle se soit modifiée lorsque la descente fictive y accède, probablement plusieurs minutes plus tard; on prévient les conséquences fâcheuses d'une telle situation en mettant, dans chaque entrée du MMS, un "numéro de modification" de la page lue en descente réelle, numéro incrémenté lors de toute mise à jour de la page dynamique. Toute divergence entre numéros de modification lu sur une page lors de la descente fictive d'une part, et gardé dans l'entrée correspondante du MMS d'autre part, produit une interruption de la descente fictive et demande que l'opérateur confirme son choix.

En conclusion, par la correction de message selon les deux modes que nous venons de décrire, ou par l'enchaînement de transactions au moyen des pages clef, on s'efforce, dans l'arbre de capture Diogène, d'épargner à l'opérateur la répétition fastidieuse des portions réutilisables du message de transaction, tout en préservant, grâce à la descente fictive, le plus de flexibilité possible dans la conception des pages et dans les interactions avec les chaînes d'application.

2. Aspects théoriques

2.1. Economie de la prise de messages

Le début du chapitre 1 a mis en évidence le fait que, dans un système d'information hospitalier, la prise des messages de transaction doit être "économique". Dans ce contexte, l'économie est tout ce qui a trait à la minimisation des ressources qu'implique la prise des messages :

- le temps et la peine des personnes ayant affaire au SIH en tant qu'utilisateurs;
- le matériel et le temps machine dévolus à la constitution des messages;
- l'effort de développement de l'ensemble des composants du logiciel servant à la prise et à la validation des messages (interface entre monde extérieur et programmes d'application proprement dits).

C'est surtout du premier de ces aspects qu'il sera question, (le troisième est évoqué au paragraphe 3.1 et le second, au paragraphe 4.4.2); non seulement parce que les utilisateurs importent avant tout, mais en raison des difficultés qu'offrent les deux autres points, particulièrement le dernier, lorsqu'il s'agit d'établir des comparaisons entre systèmes développés dans des circonstances le plus souvent fort différentes.

Les paragraphes qui suivent résument d'abord les arguments donnés en général en faveur de la prise de messages par arbre de capture, puis tentent de dégager les raisons fondamentales de l'économie réalisée par cette technique.

2.1.1 Economie de la prise de messages par arbre de capture en milieu hospitalier selon Le Beux et al.

L'article [Le Beux et al. 74] consacré à un AC développé à l'Université de Californie (San Francisco Medical Center) contient une analyse intéressante de l'économie de prise de messages par AC dont voici l'essentiel :

- a) Les messages de transactions doivent être passés au système avec précision et rapidité, moyennant le minimum d'effort.
- b) Or les erreurs avérées ou seulement potentielles entraînent un gaspillage de ressources, à savoir :
 - temps et concentration supérieurs de la part du donneur d'ordre pour introduire son message même s'il est correct d'emblée;
 - temps machine pour vérifier que le message est correct ou diagnostiquer les erreurs;

- effort d'analyse et de programmation liés à cette vérification du message et au diagnostic des erreurs;
 - temps perdu à recomposer et révérier le message s'il était erroné. Ces derniers points concernent surtout les messages d'une certaine complexité.
- c) La prise de messages par le moyen conventionnel du clavier alphanumérique prête à trois catégories d'erreurs :
- 1) erreurs lexicales : dues à l'inattention ou à l'ignorance de l'orthographe; naturellement le jargon médico-pharmaceutique et les noms propres portent à ces erreurs.
 - 2) erreurs syntaxiques : résultant d'infractions à des règles plus ou moins rigides quant à la ponctuation et à l'ordre des termes dans le message; erreurs difficilement évitables de la part de personnels médical et infirmier peu friands d'exercices informatiques.
 - 3) erreurs sémantiques : résultant, par exemple, du fait qu'un message semble correct mais que le donneur d'ordre et le système lui donnent un sens différent; ce sont les erreurs les plus pernicieuses.
- d) Par AC les erreurs lexicales et syntaxiques sont quasiment éliminées.

En effet le donneur d'ordre sélectionne des informations pré-orthographiées, et le message, construit par concaténation des informations sélectionnées, peut être, par la concaténation elle-même, pourvu de la ponctuation et des délimiteurs voulus. Quant à l'ordre des termes il est imposé par l'enchaînement des affichages. L'AC permet de prévenir, dans une large mesure, les erreurs sémantiques; chaque affichage peut être assorti de commentaires et mises en garde précisant au donneur d'ordre la signification exacte de son choix; les risques de malentendus sont fortement réduits.

2.1.2 Apport éventuel des formulaires affichables

L'analyse précédente oppose l'AC à l'entrée de messages par un clavier alphanumérique qu'on suppose être celui d'un télétype avec sortie papier. L'avantage de l'AC subsiste-t-il si le clavier alphanumérique équipe un terminal pourvu d'un écran permettant l'affichage de formules à remplir où l'on peut définir, par programme, des champs de taille et d'emplacement déterminés, protégés ou non, transmissibles ou non, présentés de diverses manières (noir sur blanc ou blanc sur noir, atténué ou normal, etc.), toutes facilités qui simplifient le remplissage de la formule ?

Deux cas extrêmes peuvent se présenter :

- a) le remplissage des formules consiste à cocher certaines rubriques; de plus, l'enchaînement des formules et le contenu de chacune d'elles sont tels que seules les rubriques pertinentes, compte tenu de ce qui a été entré sur les formules antérieures, peuvent être cochées sur chaque formule : on a affaire à une présentation particulière d'AC;
- b) à chaque type de transaction est associée une formule de présentation adéquate dont les rubriques, à remplir par le donneur d'ordre, sont du genre : nom de votre unité, votre nom, nom et prénom du patient, nom du médicament, dose désirée, etc. Une telle solution est évidemment très proche du télétype à sortie papier, avec un avantage quant au taux d'erreurs syntaxiques et sémantiques.

Conclusion : la formule à remplir tend, au mieux, en fait d'économie, vers l'AC.

Il reste que dans un projet comme Diogène ou celui auquel se rapporte [Le Beux et al. 74] le formulaire affichable demeure indispensable pour introduire certaines portions de messages dans les deux situations caractéristiques suivantes :

- 1) entrée d'informations vraiment nouvelles pour le système, qu'il n'est donc pas possible de valider entièrement par comparaison avec des informations préexistantes dans le système (banque de données ou programmes d'application); exemple : le nom d'un patient entrant pour la première fois en contact avec le système. De telles informations seront désignées par l'expression "message en format libre authentique" ou MFL authentique.
- 2) entrée d'informations qui, tout en étant validables relativement à la banque de données existante, sont difficilement produites par sélections successives; exemple : le nom d'un médicament si le donneur d'ordre est une infirmière; sa formation ne lui permet pas de spécifier un médicament suivant une classification dont la méthode n'apparaît qu'à un médecin; une telle portion de message sera dite "pseudo-MFL".

Ces deux situations ne peuvent évidemment pas être négligées dans la réalisation pratique d'une interface de prise de messages; aussi bien les pages type-C jouent-elles un rôle important dans l'arbre de capture Diogène. Mais au niveau relativement général de cette étude comparative, et compte tenu du fait que, pour ces portions de message, le taux d'erreur (donc l'économie) dépend assez peu de la méthode utilisée, nous pouvons faire abstraction de ces pages.

2.1.3 Une analyse sémiologique

Pour affirmer de manière plus générale le caractère optimum de la technique de l'AC, il convient de mieux dégager les fondements de l'économie qu'elle réalise. Celle-ci s'apparente aux mécanismes généraux de l'économie découverts initialement en linguistique générale et étendus, par la sémiologie, aux processus de communication les plus divers. André Martinet a développé en linguistique la théorie dite de la double articulation et mis en évidence le rôle essentiel que celle-ci joue en économie du langage. Citons ici [Martinet 70], extraits des paragraphes 1-8 et 1-9 * :

"La première articulation du langage est celle selon laquelle tout fait d'expérience à transmettre, tout besoin qu'on désire faire connaître à autrui s'analysent en une suite d'unités douées chacune d'une forme vocale et d'un sens. Si je souffre de douleurs à la tête, je puis manifester la chose par des cris. Ceux-ci peuvent être involontaires; dans ce cas ils relèvent de la physiologie. Ils peuvent aussi être plus ou moins voulus et destinés à faire connaître mes souffrances à mon entourage. Mais cela ne suffit pas à en faire une communication linguistique. Chaque cri est inanalysable et correspond à l'ensemble, inanalysé, de la sensation douloureuse. Tout autre est la situation si je prononce la phrase j'ai mal à la tête. Ici il n'est aucune des six unités successives j', ai, mal, à, la, tête qui corresponde à ce que ma douleur a de spécifique. Chacune d'entre elles peut se retrouver dans de tout autres contextes pour communiquer d'autres faits d'expérience : mal, par exemple, dans il fait le mal, et tête dans il s'est mis à leur tête. On aperçoit ce que représente d'économie cette première articulation : on pourrait supposer un système de communication où, à une situation déterminée, à un fait d'expérience donné correspondrait un cri particulier. Mais il suffit de songer à l'infinie variété de ces situations et de ces faits d'expérience pour comprendre que, si un tel système devait rendre les mêmes services que nos langues, il devrait comporter un nombre de signes distincts si considérable que la mémoire de l'homme ne pourrait les emmagasiner. Quelques milliers d'unités, comme tête, mal, ai, la, largement combinables, nous permettent de communiquer plus de choses que ne pourraient le faire des millions de cris inarticulés différents." (...)

"Chacune de ces unités de première articulation présente, nous l'avons vu, un sens et une forme vocale (ou phonique). Elle ne saurait être analysée en unités successives plus petites douées de sens : l'ensemble tête veut dire "tête" et l'on ne peut attribuer à tê- et à -te des sens distincts dont la somme serait équivalente à "tête". Mais la forme vocale est, elle, analysable en une succession d'unités

*Ces extraits sont reproduits avec l'aimable autorisation de la Librairie Armand Colin à Paris.

dont chacune contribue à distinguer tête, par exemple, d'autres unités comme bête, tante ou terre. C'est ce qu'on désignera comme la deuxième articulation du langage. Dans le cas de tête, ces unités sont au nombre de trois; nous pouvons les représenter au moyen des lettres t e t, placées par convention entre barres obliques, donc /tet/. On aperçoit ce que représente d'économie cette seconde articulation : si nous devons faire correspondre à chaque unité significative minima une production vocale spécifique et inanalysable, il nous faudrait en distinguer des milliers, ce qui serait incompatible avec les latitudes articulatoires et la sensibilité auditive de l'être humain. Grâce à la seconde articulation, les langues peuvent se contenter de quelques dizaines de productions phoniques distinctes que l'on combine pour obtenir la forme vocale des unités de première articulation." (...)

"Un énoncé comme j'ai mal à la tête ou une partie d'un tel énoncé qui fait un sens, comme j'ai mal ou mal, s'appelle un signe linguistique. Tout signe linguistique comporte un signifié, qui est son sens ou sa valeur, (...) et un signifiant grâce à quoi le signe se manifeste. (...) Il n'existe pas de terme universellement admis pour désigner les unités de la première articulation. Nous emploierons ici celui de monème.

"Comme tout signe, le monème est une unité à deux faces, une face signifiée, son sens ou sa valeur, et une face signifiante qui la manifeste sous sa forme phonique et qui est composée d'unités de deuxième articulation. Ces dernières sont nommées des phonèmes."

La théorie de la double articulation ne s'applique pas qu'au langage. Prieto (voir par exemple [Prieto 75], pages 32 et suivantes) a montré, qu'appliquée à d'autres situations sémiologiques, cette théorie y fait apparaître les mêmes avantages en fait d'économie de la communication. Cette démarche, dans le cas de la prise de messages de transaction, consiste par exemple à étudier la relation sémiologique SIH vers donneur d'ordre, définie par la correspondance :

<u>sémiologie</u>	————	<u>prise de messages</u>
émetteur	————	système transactionnel
récepteur	————	donneur d'ordre
signe	————	affichage sur console de visualisation

L'étude de cette relation, SIH vers donneur d'ordre, se justifie surtout si la prise de messages se fait par formulaires affichés; il s'agit alors de déterminer les dispositions d'affichages offrant les meilleures chances de réussite du remplissage correct de la formule par le donneur d'ordre.

La relation en sens inverse, donneur d'ordre vers SIH, appliquée globalement à la prise d'un message de transaction, offre plus d'intérêt pour l'arbre de capture; cette relation met en jeu la correspondance suivante :

<u>sémiologie</u>	————	<u>prise de messages</u>
émetteur	————	donneur d'ordre
récepteur	————	système transactionnel
signe ou sème	————	message de transaction

Explicitons les caractéristiques du code supportant cette relation. Abstraction faite (par la remarque de la fin du paragraphe 2.1.2) de portions de message non validables par rapport à la banque de données, le nombre de messages de transaction différents acceptables par le système à un moment donné, est fini, compris entre 10^{10} et 10^{20} , mettons que ce soit 10^{15} . Cette évaluation inévitablement grossière, se fonde sur un examen d'arbres comme ceux qui servent à la démonstration du paragraphe 1.2, pour déterminer le nombre moyen d'affichages produisant un message complet et, sur chaque affichage, le nombre moyen de sélections non masquées. Avec cette valeur de 10^{15} la nécessité d'une articulation du code apparaît immédiatement; reprenant le texte cité de Martinet, on conçoit mal en effet, d'associer à chacun des faits d'expérience à traduire, un cri ou un dessin différents, en admettant que le système s'accommode de l'un ou de l'autre. D'où l'idée d'utiliser un générateur d'unités de seconde articulation, par exemple un clavier alphanumérique. Combien d'unités distinctes doit pouvoir produire ce générateur ? Il semble, si l'on se réfère au nombre de phonèmes du langage parlé (compris selon les langues entre vingt et trente), ou au nombre de signes distincts des alphabets usuels, que le compromis à trouver entre variétés des unités et agilité dans leur utilisation, conduise à un nombre d'unités distinctes de l'ordre de quelques dizaines; partons de vingt pour fixer les idées. Il suffirait donc de suites d'environ 10 unités de seconde articulation pour représenter les 10^{15} messages possibles.

De telles chaînes d'unités de seconde articulation pourraient éventuellement servir de clefs d'accès à un fichier gigantesque contenant les 10^{15} messages possibles préfabriqués; cela n'est guère réaliste mais offre quelque intérêt théorique comme on le verra plus loin. Il est sûr que le donneur d'ordre ne peut générer une clef sous cette forme (du moins pas sans artifice, artifice que permet précisément l'arbre de capture); ceci vient de ce que les unités de seconde articulation n'ont, par définition, pas de sens.

C'est pourquoi, dans les systèmes n'utilisant pas l'arbre de capture, le donneur d'ordre va produire, par le générateur d'unités de seconde articulation (en général un clavier alphanumérique), le texte des messages de transaction en langage plus ou moins naturel. Ce texte compte en moyenne peut-être 150 unités de seconde articulation; s'il est entré au hasard il y a en gros 20^{150} , soit environ 10^{200} messages possibles: le gaspillage et le risque d'erreurs sont considérables. On objectera, non sans raison, que quel que soit le confort que l'on entend offrir au donneur d'ordre, on peut attendre de sa part, lors de l'entrée d'un message de transaction, une concentration qui hisse ses performances sensiblement au-dessus de celles d'un chimpanzé au clavier d'une machine à écrire. Toutefois, même les textes corrects en langage vraiment naturel, et de longueur équivalente, offrent, par rapport à l'ensemble des messages de transaction acceptables, le spectacle d'un inimaginable foisonnement; on peut admettre, en effet [Shannon 65] que, compte tenu de toutes les redondances de la langue anglaise, par exemple, l'apport d'information par lettre d'un texte naturel anglais est de 1 bit (au lieu de $\log_2 27$, 27 et non 26 à cause de l'espace considéré comme un caractère). Dans cette hypothèse, avec 150 caractères on peut générer 2^{150} soit environ 10^{50} phrases faisant un sens en une langue naturelle imposée.

La rareté relative des messages de transaction acceptables provient

- de la rareté relative des monèmes autorisés dans ces messages considérés globalement (vocabulaire médical et termes relatifs à la fixation d'un rendez-vous, essentiellement);
- du fait qu'en un endroit d'un message donné, le nombre de monèmes effectivement substituables, eu égard à la sémantique et à la syntaxe, est extrêmement réduit.

Or toutes ces limitations sont implicitement contenues dans le système transactionnel (banque de données et/ou programmes d'application) quel que soit le mode de prise des messages, puisque le système doit être capable de trier, parmi tous les messages possibles, ceux qui sont effectivement acceptables.

L'économie de la prise de messages réalisée par arbre de capture provient justement, pour l'essentiel, de ce qu'il utilise ces limitations de façon "positive" ou "générationnelle", en tant que règles de production de messages valides par construction, plutôt que de manière "négative" ou "répressive", après que le message a été produit par ailleurs, pour écarter, des quelques messages acceptables, la masse énorme des messages incorrects possibles.

Utiliser positivement ces limitations, autrement dit définir un arbre de capture, revient à :

1. Répertorier les unités de première articulation ou monèmes possibles, c'est-à-dire validables relativement à la banque de données. Ces monèmes "informatiques" groupent en général plusieurs monèmes de langue naturelle. Un trait d'économie intéressant de l'arbre Diogène est que ces monèmes informatiques ont, comme de vrais monèmes, deux faces :
 - a) face signifiante qui sera incorporée au message visible par l'opérateur (en terminologie Diogène, le Message Arbre de Capture ou MAC);
 - b) face signifiée qui sera incorporée au message exploité par les programmes d'application (Message Back-Ground, MBG).
2. Etablir, entre ces monèmes, des relations qui peuvent être de deux natures :
 - a) "paradigmatique", exprimant le fait que ces monèmes commutent entre eux en un endroit donné du message; en terminologie Diogène, ils composent les diverses sélections d'une même page ou de pages qui sont la suite les unes des autres lorsqu'une page n'offre pas assez de choix possibles;
 - b) "syntagmatique", exprimant le fait que l'apparition de tel monème doit intervenir dans un contexte défini d'ensembles commutables d'autres monèmes; en terminologie Diogène ces relations sont concrétisées par la désignation de pages "successeurs" dans les sélections.
3. Définir les règles de cheminement dans le graphe établi sous 2. Un aspect pratique important de ce cheminement est qu'une intervention simple du donneur d'ordre, assimilable à la production d'une unité de seconde articulation, suffit
 - a) à spécifier le monème choisi sur une page;
 - b) à inclure aux faces signifiantes et signifiées du message de transaction les faces correspondantes du monème sélectionné;
 - c) à susciter l'envoi, par le système au donneur d'ordre, de l'affichage suivant.

Une autre manifestation de l'économie réalisée par l'AC tient donc dans les constatations suivantes :

- il y a de la part de l'arbre un effet multiplicatif : l'unité de seconde articulation produite par l'opérateur (caractère étiquetant un choix) est convertie dans le message en unité de première articulation; c'est l'artifice dont il était question plus haut;
- la chaîne des unités de seconde articulation produites pour un message de transaction donné est un codage de ce message. Il en sera question plus loin (paragraphe 2.2);
- malgré ce codage, la face signifiante du message a les apparences du langage naturel, en particulier la redondance qui en détermine la lisibilité.

Il convient d'insister sur le fait que si la sélection d'un monème n'implique que la production de ce qui, en l'absence d'AC, serait une unité de seconde articulation du code supportant la relation donneur d'ordre / système, ceci ne signifie pas que le message de transaction produit par AC présente cette seconde articulation. Il faut y voir essentiellement un trait d'économie de la réponse à la relation mentionnée au début de ce paragraphe (avec émetteur = système, récepteur = donneur d'ordre, signe = affichage).

2.1.4 Résumé et conclusion de l'analyse sémiologique

Les deux raisons suivantes sont à la base de l'économie réalisée par l'AC :

- utilisation générative des limitations imposées aux messages de transaction acceptables par le système transactionnel. Dans l'AC ces limitations se présentent sous forme d'un graphe où les chemins permis ne produisent que des messages valides;
- effet multiplicatif sur l'information entrée par le donneur d'ordre; la production d'une unité de seconde articulation génère une contribution d'un monème au niveau du message de transaction.

Ces deux raisons rendent l'AC avantageux alors même que le coût machine total de production d'une unité de seconde articulation dans l'arbre est supérieur à celui d'une unité de seconde articulation conventionnelle, puisque toute sélection implique en général l'envoi d'un affichage complet dans un temps nécessairement court (de l'ordre de la seconde) pour rendre cette technique acceptable en pratique.

Pour éviter tout malentendu il faut rappeler que dans les messages de transaction envisagés ici, les portions de message en format libre sont ignorées; l'effet multiplicatif de l'arbre ne pourrait s'y manifester et son usage reviendrait à produire de simples unités de seconde articulation par un moyen plus coûteux que le clavier alphanumérique. Le caractère optimal de l'AC n'est donc établi que dans les cas où son emploi n'est pas manifestement inadéquat. D'ailleurs si l'analyse précédente a permis de mieux cerner les raisons de l'économie réalisée par l'AC elle n'a pas donné la preuve de son optimalité. Le paragraphe suivant comble cette lacune.

2.2. Recherche d'un arbre de capture optimum

Ce paragraphe prolonge, sur des bases apparemment moins subjectives, l'étude "économique" de la prise de messages abordée au paragraphe précédent. D'étroites analogies avec la théorie du codage de l'information et celle des questionnaires, suggèrent l'esquisse de méthodes propres à déterminer l'AC optimum pour la prise de messages dans l'ensemble fini des messages de transaction que le système juge acceptables à un instant ou dans un état donnés. Optimum signifie ici : demandant le moins de choix successifs de la part du donneur d'ordre.

De manière plus générale, et au prix d'hypothèses relativement peu restrictives, des considérations du même ordre établissent que, parmi tous les moyens possibles de saisie des messages, l'AC permet de faire tendre effectivement vers le minimum prévu par la théorie de l'information, l'information que le donneur d'ordre doit entrer dans le système pour caractériser un message.

Pour faciliter la lecture de ce paragraphe, les notations adoptées sont énumérées en 2.2.4.

2.2.1 Formulation du problème d'optimisation

Le principe de la prise de messages par AC est provisoirement admis. On suppose donnés l'ensemble fini

$$\mathcal{M} = \{m_i \mid i=1 \dots N\}$$

des messages acceptables par le système dans un état particulier (autrement dit, à un instant donné de la vie de l'Hôpital), ainsi que la suite $p_1, \dots, p_N$ des probabilités qu'ont $m_1, \dots, m_N$ respectivement d'intervenir en cet état. Si n_i désigne le nombre de sélections nécessaires pour produire m_i , le problème posé est alors de construire un arbre de capture minimisant la moyenne

$$n = \sum_{i=1}^N n_i p_i$$

du nombre de sélections requis pour générer un message pris au hasard dans $\mathcal{M}$.

Il faut préciser encore que les pages de type-C ne sont pas envisagées à ce stade de l'analyse. En outre deux grandeurs sont considérées comme fixes dans le processus d'optimisation :

- le nombre maximum de choix à offrir sur une page de type-S est imposé;
- le "coût" de chaque sélection est constant; le coût de tout message est donc proportionnel au nombre de sélections qu'il exige pour être produit, indépendamment de la forme prise par l'AC du fait de l'optimisation.

Il existe plusieurs formes transposées de ce problème; toutes reviennent à la recherche d'une arborescence dont la longueur de cheminement (espérance mathématique de la longueur des chemins unissant la racine aux sommets terminaux) est minimale. Dans le cas de l'AC, l'arborescence considérée est l'arborescence externe définie au paragraphe 1.3. Le mode effectif d'enchaînement des affichages n'intervient pas ici, si ce n'est par les contraintes qu'impose le contenu sémantique des choix à proposer sur les affichages; l'optimisation doit en effet préserver une hiérarchie de concepts guidant l'opérateur de façon naturelle vers le message à produire, faute de quoi l'AC perdrait sa justification essentielle; un exemple clarifiera ce point en temps voulu.

2.2.2 Reformulations du problème

2.2.2.1 Codage optimum des messages de transaction

Soit un alphabet fini

$$\mathcal{A} = \{c_j \mid j=1\dots a\}$$

de a signes ou caractères distincts. Une représentation d'un message $m_i \in \mathcal{M}$ dans l'alphabet $\mathcal{A}$ est une chaîne r_i de ℓ_i signes pris (avec ou sans répétitions) dans $\mathcal{A}$. Un codage de l'ensemble $\mathcal{M}$ des messages dans l'alphabet $\mathcal{A}$ est une bijection de $\mathcal{M}$ sur un sous-ensemble de $\mathcal{A}^*$, ensemble infini dénombrable des chaînes que l'on peut former avec des caractères de $\mathcal{A}$.

Un codage optimum de $\mathcal{M}$ dans l'alphabet $\mathcal{A}$ minimise la taille moyenne ℓ des chaînes du sous-ensemble image de $\mathcal{M}$, dont l'expression est

$$(2.1) \quad \ell = \sum_{i=1}^N p_i \ell_i .$$

L'identité du problème de la recherche du codage optimum et du problème initial posé apparaît en assimilant l'alphabet $\mathcal{A}$ à l'ensemble des caractères étiquetant les choix possibles sur une page de type-S dans l'AC.

Sous une forme proche de celle qui vient d'en être donnée, le problème du codage optimum a servi de point de départ aux travaux de Shannon sur la théorie de l'information Shannon 48, Khinchin 57, qui ont montré que le minimum possible de la taille moyenne des chaînes représentatives de $\mathcal{M}$ est

$$(2.2) \quad \ell_{\min} = - \sum_{i=1}^N p_i \log_a p_i = \sum_{i=1}^N p_i \log_a \frac{1}{p_i},$$

où p_i est comme précédemment la probabilité d'occurrence du message $m_i \in \mathcal{M}$. Cette valeur de $\ell_{\min}$ n'est effectivement réalisable que si les p_i sont des puissances entières négatives de a . Le problème posé est en effet non seulement de connaître la taille minimum des chaînes mais de produire ce codage optimum. L'algorithme de Huffman indiqué plus loin permet de construire les chaînes réalisant la meilleure approximation de $\ell_{\min}$ pour un jeu donné des p_i . L'expression (2.2) de $\ell_{\min}$ définit aussi la quantité d'information I apportée par un message pris au hasard dans l'ensemble $\mathcal{M}$ suivant la distribution p_i . La base a du logarithme indique que l'information est mesurée en unités "caractères de $\mathcal{A}$ ". Lorsque $a=2$, I est mesurée en "bit". Il est inutile de reprendre et discuter ici les hypothèses et développements conduisant à (2.2) et légitimant la définition ainsi donnée de la quantité d'information; les références [Khinchin 57] et [Picard 72] fournissent tous les détails, et l'examen d'un cas particulier rend ces résultats plausibles. Un rapprochement de (2.1) et (2.2) montre que la taille individuelle des chaînes devrait être $\log_a(1/p_i)$; la présence du logarithme s'explique immédiatement si le nombre de messages N est une puissance entière k de a : $N=ak$, et que tous les messages sont équiprobables. Par symétrie les chaînes ont toutes la même taille, et l'univocité de la représentation implique que ces chaînes aient $k=\log_a N = \log_a(1/p)$ caractères au minimum. En outre, l'information apportée par un tel message est celle de k décisions successives prises parmi a possibilités offertes à chaque étape, donc de k unités lorsque l'unité d'information adoptée est "un choix parmi a possibilités"; or (2.2) donne bien dans ce cas

$$I = \ell_{\min} = \sum_{i=1}^N p \log_a(1/p) = N \frac{1}{N} k = k.$$

Le rapport établi précédemment entre codage des messages d'une part, et AC d'autre part, a pour conséquence remarquable la preuve du caractère optimum de l'arbre de capture dans l'absolu. En effet, du point de vue du donneur d'ordre, le coût de production d'un caractère de l'alphabet $\mathcal{A}$ (ensemble des caractères étiquetant les choix offerts sur les pages type-S) est du même ordre que celui résultant pour lui de l'usage de tout autre alphabet, par un clavier de machine à écrire notamment. Certes l'AC demande-t-il

que l'opérateur lise une page et attende l'affichage suivant; mais avec un peu d'habitude, et si le temps de réponse est bon, cet effort supporte avantageusement la comparaison avec l'épuisement que provoque l'hésitation sur l'orthographe des noms, une date, la valeur d'un code ou la syntaxe du message. Dès lors, l'identité relevée entre optimisation du codage des messages et optimisation de l'AC, établit que, du point de vue du donneur d'ordre, il ne peut être trouvé de moyen de saisie des messages plus économique qu'un AC optimisé.

2.2.2.2 L'arbre de capture en tant que questionnaire

Pour aborder la seconde reformulation du problème d'optimisation, on observe qu'un message particulier est la synthèse des réponses fournies à une suite de questions visant à préciser les intentions du donneur d'ordre dans tout le champ de ce que le système est susceptible d'accepter comme messages de transaction, dans un état donné. Vu sous cet angle, l'AC est un "questionnaire" à choix multiples ayant pour support l'arborescence de l'arbre. C.F. Picard a élaboré une théorie de tels questionnaires [Picard 72]. Un questionnaire Q est défini par un triplet $Q=(X, \Gamma, P_\Gamma)$, où les deux premiers éléments (X, Γ) forment un graphe, le support de Q , dont X est l'ensemble des sommets et Γ celui des arcs. P_Γ est une probabilité attachée aux arcs de Γ .

Pour être effectivement un questionnaire au sens de Picard, le graphe valué $Q=(X, \Gamma, P_\Gamma)$ doit encore jouir des deux propriétés suivantes :

1. Son support (X, Γ) est un "latticiel", c'est-à-dire une arborescence dans laquelle on autoriserait les cycles (plusieurs arcs peuvent converger vers un même sommet). Le schéma ci-dessous clarifie ces nuances :

arborescence



latticiel



Dans l'ensemble X des sommets du support on distingue l'ensemble E des N sommets terminaux ou réponses et l'ensemble F des sommets intérieurs ou questions.

2. La valuation P_{Γ} a les caractéristiques d'une probabilité. Une loi de Kirchhoff traduit sa conservation à tout sommet $i \in F$:

$$\sum_{j \in \Gamma_i} p(i, j) = \sum_{h \in \Gamma_i^{-1}} p(h, i)$$

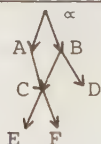
où Γ_i^{-1} et Γ_i désignent respectivement l'ensemble des arcs entrant et sortant en i ; $p(i, j)$ est la probabilité attachée à l'arc (i, j) . Sur l'ensemble des sommets terminaux, le fait que P_{Γ} soit une probabilité impose que

$$\sum_{i \in E} \sum_{h \in \Gamma_i^{-1}} p(h, i) = 1.$$

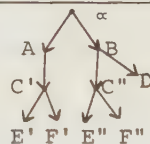
Cette dernière condition est assez accessoire; tous les résultats de la théorie des questionnaires demeurent valables si l'on interprète la valuation P_{Γ} non comme une probabilité mais comme un "coût" ou une "importance" par exemple, auxquels cas la normalisation n'offre en général pas d'intérêt.

Un questionnaire est optimal lorsque sa longueur de cheminement est minimale parmi tous les questionnaires d'une famille admettant le même ensemble de réponses de probabilités données et la même base (nombre maximum admissible d'arcs sortant d'un sommet). Les méthodes d'optimisation connues ne peuvent s'appliquer directement qu'aux questionnaires arborescents; toutefois, de tout questionnaire latticiel non arborescent il est possible de déduire un questionnaire arborescent compatible dont le support est l'arborescence des chemins du latticiel support du questionnaire original; la valuation du questionnaire arborescent compatible est dérivée de la valuation originale par des règles de proportionnalité dont le détail n'importe guère. Le schéma suivant indique par contre comment se construit l'arborescence des chemins d'un latticiel :

latticiel



arborescence des chemins



Or l'arborescence externe d'un arbre de capture joue relativement à un graphe rendant compte du mode d'enchaînement des affichages au cours d'une descente (voir paragraphe 2.3), un rôle comparable à celui du questionnaire arborescent compatible vis-à-vis du questionnaire latticiel dont il dérive. Cependant l'AC et le questionnaire au sens de Picard

différent de manière remarquable, sur le plan informationnel, dans leur relation avec les arborescences externe et compatible associées. L'information $I(A)$ transmise par l'arbre de capture A est cumulée sous forme de message au cours de la descente, d'où son expression

$$I(A) = - \sum_{i=1}^N p_i \log_a p_i$$

où p_i est la probabilité du message m_i . L'information $I(Q)$ transmise par le questionnaire Q est au contraire concentrée dans les sommets terminaux (réponses) :

$$I(Q) = - \sum_{y \in E} p(y) \log p(y)$$

où E est l'ensemble des réponses du questionnaire [Picard 72, p. 77]. Comme il y a bijection de l'ensemble $\mathcal{M}$ des messages sur l'ensemble des chemins (donc des sommets terminaux) de l'arborescence externe de A , $I(A)$ se confond avec l'information transmise par le questionnaire arborescent que constitue l'arborescence externe de A . Par contre l'information $I(Q)$ transmise par le questionnaire latticiel Q est inférieure à celle que transmet un questionnaire arborescent compatible à Q . D'où la distinction introduite par Picard dans un questionnaire latticiel entre informations transmises I , traitées J et absorbées K , liées par $J(Q) = I(Q) + K(Q)$. K , nulle si le questionnaire est arborescent, résulte de l'absorption d'information aux sommets du latticiel où entrent plus d'un arc.

L'importance de la distinction faite à cet égard entre AC et questionnaire est mise en évidence par la constatation suivante : à la limite, un questionnaire de support quasi-fortement connexe (dont toutes les réponses sont confondues) ne transmet aucune information [Picard 72, p. 81]; toute l'information traitée dans la partie divergente du graphe est absorbée dans sa partie convergente, ce que laisse bien présumer le fait qu'un tel questionnaire n'établit aucune discrimination. Si l'AC était simplement assimilé à un questionnaire latticiel dont le questionnaire arborescent compatible serait l'arborescence externe on conclurait, de manière évidemment erronée, que l'arbre de capture Diogène ne transmet aucune information puisque toute descente se termine en principe par le même choix "OK" sur l'affichage EXAC2 (chapitre 1).

En résumé :

1. Du point de vue informationnel l'AC a les propriétés d'un questionnaire arborescent dont le support est l'arborescence externe de l'AC.

2. Le problème initial de l'optimisation d'un AC se ramène donc à celui de l'optimisation d'un questionnaire arborescent.
3. Le mode effectif d'enchaînement des affichages présentés successivement n'intervient pas à ce stade; en particulier il ne faut pas assimiler l'AC à un questionnaire latticiel en se fondant sur l'identité des affichages présentés en divers sommets de l'arborescence externe.

2.2.2.3 Arbre de capture et recherche d'information par une arborescence optimale

Pour cette troisième reformulation du problème initial, l'arborescence externe de l'AC est assimilée à celle qu'on est amené à construire dans certains types de recherche optimisée d'informations dans une liste ordonnée. Cette méthode est présentée par exemple dans [Knuth 73, §§ 6.2.2-4]: il s'agit, dans le contexte de la création d'un index KWIC, d'écarter, en tant qu'entrées de l'index, tous les mots appartenant à une liste de mots triviaux (antidictionnaire); ces mots, rangés par ordre alphabétique, forment les sommets d'une arborescence binaire qui doit être optimale de manière à minimiser le nombre de comparaisons nécessaires entre mot candidat à l'index et mots de l'antidictionnaire, compte tenu de la probabilité pour un mot candidat, pris au hasard dans le corpus à indexer, d'être tel mot particulier de l'antidictionnaire (probabilité dite intérieure) ou de se trouver dans l'intervalle (au sens de l'ordre lexicographique) entre telle paire de mots consécutifs de l'antidictionnaire (probabilité dite extérieure).

On peut voir que l'arborescence externe de l'AC est assimilable à une arborescence de recherche où les probabilités intérieures sont nulles :

- les comparaisons avec les mots de l'antidictionnaire correspondent aux sélections sur les pages successives;
- les intervalles entre mots consécutifs de l'antidictionnaire correspondent aux messages produits par descente dans l'AC.

On suppose ainsi que tous les messages acceptables par le SIH dans un état donné, préexistent en tant qu'articles d'un fichier immense, évidemment imaginaire; l'arborescence externe de l'AC est alors l'arborescence de recherche (qu'il s'agit d'optimiser) d'un message ou article particulier de ce fichier imaginaire.

2.2.3 Possibilités d'optimisation

Les méthodes d'optimisation d'arborescences que l'on trouve proposées dans les domaines où le problème initial a été reformulé, relèvent de trois classes. Il s'agit de constructions optimales :

- a. sans contraintes;
- b. avec préservation d'un ordre total des sommets terminaux, ordre qui pourrait, par exemple, refléter un embryon de sémantique;
- c. avec préservation de partitions des sommets terminaux, partitions traduisant une sémantique : elles interdisent certains des regroupements de sommets auxquels conduirait une optimisation réalisée sans contrainte.

Il convient d'observer d'emblée que le volume de l'arborescence externe de l'AC Diogène, comme la difficulté d'exprimer les contraintes sémantiques à respecter, interdisent toute optimisation globale de cet arbre, ce qui n'empêche pas les principes développés ici de s'appliquer à des parties de l'arbre.

2.2.3.1 Optimisation sans contrainte, algorithme de Huffman

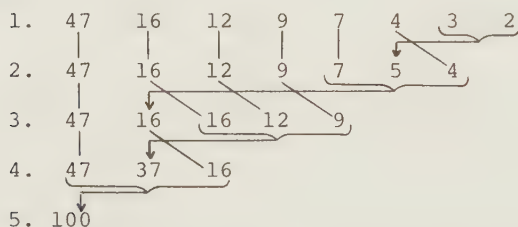
Conçu à l'origine pour le codage binaire optimum de textes [Huffman 52], autrement dit la construction d'arborescences binaires homogènes optimales, cet algorithme se généralise au cas d'une arborescence homogène de base quelconque > 2 [Knuth 68, p. 405], ou à celui d'une arborescence hétérogène où le nombre de sommets permis dans chacune des bases données est imposé [Picard 72, p. 50]; dans ce contexte, la "base" désigne toujours le nombre d'arcs sortant d'un sommet non terminal d'une arborescence ou d'un latticiel. Une arborescence construite selon l'algorithme de Huffman satisfait à la relation (2.2) ci-dessus d'autant plus qu'il est possible comme le montrent les références données. Le principe de cet algorithme ne sera ici qu'esquissé dans le contexte de l'arborescence externe, et un exemple fera ressortir la nécessité d'introduire des contraintes sémantiques.

L'idée générale est de faire en sorte que les messages les plus fréquents demandent le moins de sélections; à cet effet on part des N messages à produire rangés par ordre de probabilités croissantes. Supposant qu'il s'agisse de construire une arborescence homogène de base a , on regroupe sur une même page les a messages les moins probables, en un sommet affecté d'une probabilité égale à la somme de celle des messages regroupés. Dans la nouvelle énumération de $N-a+1$ probabilités, on regroupe à nouveau en un sommet (ou page) les a probabilités les plus faibles, et ainsi de suite, jusqu'à ce que l'énumération se réduise à a sommets regroupés en un qui est la racine; chaque regroupement définit un sommet pourvu de a arcs sortants dans

l'arborescence construite. Le rang d'un message dans l'arborescence finale, donc le nombre de sélections nécessaires pour le produire, est aussi le nombre de regroupements où ce message est intervenu, une première fois explicitement, ensuite en tant qu'affluent d'un regroupement antérieur. Comme le processus appelle en priorité les messages de probabilité faible, leur rang s'accroît relativement à celui des messages de probabilité élevée. Si, contrairement à ce que supposait la forme donnée ici à l'algorithme, $N-1$ n'est pas multiple de $a-1$, le premier regroupement porte sur un nombre réduit de messages, de manière que tous les autres sommets, la racine en particulier, admettent a arcs sortants.

Exemple : construction d'une arborescence de Huffman de base $a=3$ sur 8 messages $m_1, \dots, m_8$ de probabilités respectives (multipliées par 100) 47, 16, 12, 9, 7, 4, 3, 2.

Les regroupements effectués peuvent se schématiser comme suit :



L'arborescence ainsi construite est représentée par la figure 2-1. Le codage résultant des messages s'obtient en étiquetant les arcs issus de chaque sommet. Soit par exemple $\mathcal{A} = \{a, b, c\}$ l'alphabet adopté, la représentation des messages devient $r_1=a$, $r_2=ba, \dots$, $r_8=cbb$.

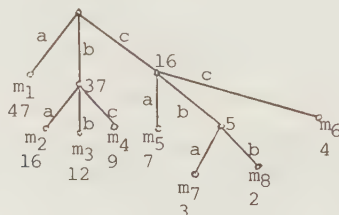


Fig. 2-1 Arborescence de Huffman construite sur 8 messages de probabilités (47, 16, 12, 9, 7, 4, 3, 2)/100. Base 3. Incertitudes de classement levées de manière à réaliser une hauteur minimale.

Cet exemple montre que l'arborescence optimale peut ne pas être unique; le regroupement de la ligne 2 du schéma ci-dessus crée un sommet de poids 16, qui, dans le classement de la ligne 3 peut prendre place avant ou après le message m_2 du poids 16 lui aussi. Le classement adopté produit une arborescence optimale de hauteur minimale; la figure 2-2 représente l'arborescence obtenue avec un classement différent; bien que de hauteur supérieure, sa longueur de cheminement est la même.

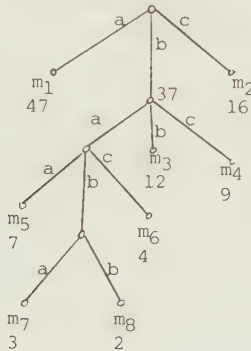


Fig. 2-2 Variante de l'arborescence de Huffman de la fig. 2-1, de hauteur non minimale.

2.2.3.2 Insuffisances sémantiques de l'optimisation huffmanienne

Donner un contenu aux huit messages de l'exemple précédent fait immédiatement apparaître les inconvénients pour l'AC d'une optimisation qui ne tiendrait compte que de la probabilité des messages; supposons que les messages aient les contenus suivants :

<u>message</u>	<u>contenu</u>	<u>probabilité</u>
m ₁	radio main	47
m ₇	radio pied	3
m ₄	radio crâne	9
m ₃	radio poumon	12
m ₅	prescription morphine	7
m ₂	prescription aspirine	16
m ₆	admission/enregistrement	4
m ₈	admission/transfert	2

Une arborescence de base $a=3$ fondée sur la sémantique aurait, pour produire ces messages, par exemple l'allure de la figure 2-3. Par contre, en portant dans l'arborescence huffmanienne de la figure 2-1 le contenu des messages qui vient d'être proposé on arrive au résultat de la figure 2-4. Cette arborescence est certes optimale mais les choix correspondant dans l'arborescence externe aux arcs entrant aux sommets A, B, C ne sont pas "formulables", selon l'expression de C.F. Picard à propos des questionnaires; en effet, dans le cas de A par exemple, il n'y a rien de commun entre "prescription aspirine" et "radio poumon" qui permette de formuler convenablement le choix représenté par l'arc OA.

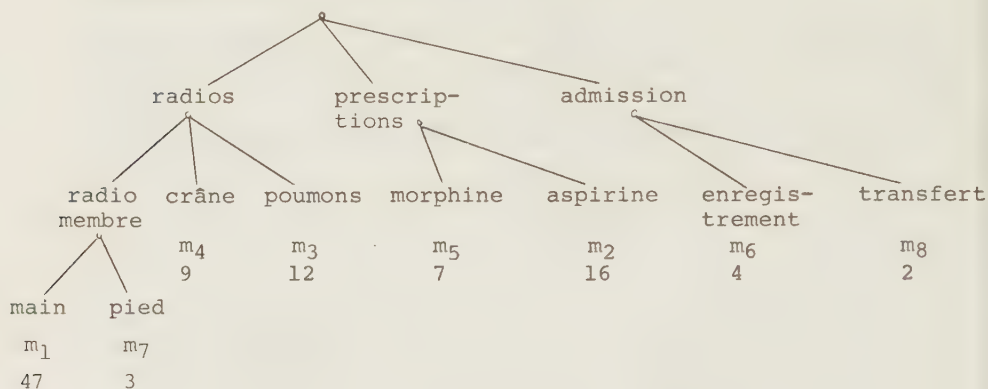


Fig. 2-3 Arborescence "sémantique" de base $a=3$, construite sur huit messages. Longueur de cheminement : 2,50.

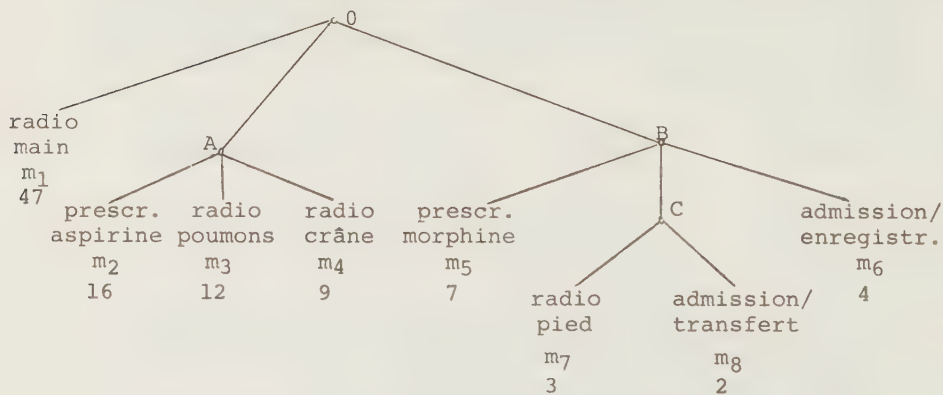


Fig. 2-4 Arborescence huffmanienne montrant le caractère informulable de certains choix. Longueur de cheminement : 1,58.

2.2.3.3 Prise en compte de la sémantique

Une comparaison des figures 2-3 et 2-4 montre que le caractère informulable de certains choix de l'arborescence optimale construite par l'algorithme de Huffman provient, en particulier, de ce que, dans cette arborescence, il n'est pas tenu compte du classement sémantique des messages donné au début du paragraphe 2.2.3.2. Au contraire, une arborescence où l'ordre des sommets terminaux coïnciderait avec un classement sémantique des messages ne laisserait subsister de choix informulables qu'aux frontières des régions sémantiques; encore est-il possible de les éviter en introduisant des messages ou sommets factices dans les zones frontière de manière à assurer une base constante sans qu'un choix ait à couvrir deux régions sémantiques voisines sans "dénominateur commun". Le problème revient donc à construire une arborescence optimale préservant un ordre total imposé aux messages. Un problème analogue est traité par [Gilbert/Moore 59] dans le contexte du codage binaire alphabétique optimal, et [Knuth 73, p. 435] dans la recherche d'information par arborescence binaire optimale. Cette voie offre deux avantages significatifs :

- La formulation des contraintes sémantiques est en principe simple; il suffit de ranger les messages dans l'ordre jugé satisfaisant du point de vue sémantique;
- le coût maximum de la contrainte que représente la préservation d'un ordre total, est connu. Gilbert et Moore ont montré que, par rapport à la longueur de cheminement donnée par (2.2) la majoration est de 2 au maximum; autrement dit, en respectant une sémantique exprimée de cette manière, on accroît de 2 au plus, le nombre moyen de sélections nécessaires pour produire un message, par rapport à un AC idéal donc les choix seraient, de toute façon en général informulables. D'ailleurs cette majoration devient négligeable si la hauteur de l'arborescence est grande.

L'algorithme de construction d'une arborescence optimale préservant un ordre total des sommets, donné par [Knuth 73] dans le cas binaire, s'étend à une base a quelconque. Le processus se déroule d'aval en amont et repose sur une condition nécessaire d'optimalité : toute sous-arborescence d'une arborescence optimale est elle-même optimale. Etant donnée la suite $p_1, \dots, p_N$ des probabilités des messages de

$$\mathcal{M} = \{m_1, \dots, m_N\}$$

où l'ordre est significatif, soit $\mathcal{M}_{ij}$ un sous-ensemble de $N_{ij} > a$ messages $m_i, \dots, m_j$ de $\mathcal{M}$, contigus au sens de l'ordre défini dans $\mathcal{M}$. Une sous-arborescence optimale couvrant $\mathcal{M}_{ij}$ (c'est-à-dire dont les sommets terminaux sont associés à

$m_1, \dots, m_j$ et préservent leur ordre) a pour racine un sommet X_{ij} tel que :

- les sous-arborescences dont les racines sont les successeurs de X_{ij} sont elles-mêmes optimales;
- ces sous-arborescences définissent une partition de l'intervalle (i, j) en $\underline{a}$ intervalles choisis de manière à minimiser la longueur de cheminement dans la sous-arborescence de racine X_{ij} , soit :

$$(2.3) \quad \ell_{ij} = \sum_{k=i}^j p_k + \text{Min} \sum_{u=1}^{\underline{a}} \ell_{i_u j_u}$$

où "Min" porte sur les $\binom{j-i-1}{\underline{a}-1}$ partitions de l'intervalle (i, j) en $\underline{a}$ intervalles :

$$(i, j) = (i_1 \dots j_1 i_2 \dots j_{u-1} i_u \dots j_u i_{u+1} \dots j_a).$$

L'expression (2.3) peut s'étendre au cas $j-i < a$ en observant que $\ell_{ii} = 0$ et que si $0 < j-i < a$, l'arborescence couvrant $\mathcal{M}_{ij}$ n'a que sa racine et les sommets terminaux $m_1, \dots, m_j$; le second terme de l'expression de ℓ_{ij} est donc nul. D'où une forme générale de la longueur de cheminement

$$(2.3') \quad \ell_{ij} = (1 - \delta_{ij}) \sum_{k=i}^j p_k + \text{Min} \sum_{u=1}^{\underline{a}} \ell_{i_u j_u}$$

dans laquelle δ_{ij} est le symbole de Kronecker (1 si $i=j$, 0 dans les autres cas). Dans les expressions (2.3) et (2.3') la longueur de cheminement de chaque sous-arborescence est pondérée par la probabilité de passage par sa racine (somme des probabilités des messages qu'elle couvre).

L'algorithme consiste à appliquer (2.3') de manière récurrente sur des intervalles (i, j) de plus en plus grands dans les limites $(1, N)$. ℓ_{1N} est la longueur de cheminement de l'arborescence recherchée; l'arborescence elle-même résulte de l'ensemble des sommets intervenant dans la partition optimale de $(1, N)$ donnant ℓ_{1N} , puis dans les partitions optimales de cette partition et ainsi de suite jusqu'aux sommets terminaux associés aux messages.

Cet algorithme appelle deux remarques :

- il fait usage de toutes les partitions de l'ensemble des messages rangés dans un ordre donné;
- en général, une classification des messages, autrement dit une suite de partitions de l'ensemble $\mathcal{M}$, exprime la sémantique de manière plus naturelle qu'un ordre défini dans $\mathcal{M}$.

Ces constatations conduisent à l'algorithme attribué à F. Dubail, dont [Picard 72, p. 138] donne l'esquisse. Il se fonde sur une relation déduite de (2.3) en y remplaçant les intervalles (i, j) et (i_u, j_u) par des classes de messages notées par exemple E_i et E_{j_u} respectivement; E_i est supposée admettre, du fait de la sémantique, des jeux de partitions en a classes E_{j_u} , $u=1, \dots, a$. La longueur de cheminement de l'arborescence optimale couvrant la classe E_i est donnée par

$$(2.4) \quad \ell_{E_i} = \sum_{m_k \in E_i} p_k + \text{Min} \sum_{u=1}^a \ell_{E_{j_u}}$$

où "Min" porte sur tous les jeux de partitions de E_i admissibles au sens de la sémantique définie sur $\mathcal{M}$. Des partitions en moins de a classes sont admises; il suffit de limiter la sommation dans le second terme; la racine de la sous-arborescence présente une base réduite en conséquence.

Comme précédemment, l'algorithme consiste à appliquer la relation (2.4) de manière récurrente à des classes de plus en plus vastes; l'algorithme s'arrête lorsque la classe E_i est l'ensemble $\mathcal{M}$ de tous les messages.

2.2.3.4 Optimisation partielle d'une arborescence existante

Une application des méthodes d'optimisation sous contraintes sémantiques à l'AC Diogène pris globalement, est exclue en raison du nombre énorme des sommets terminaux (messages différents) de son arborescence externe. Nous avons parlé de 10^{15} messages; or le temps d'exécution de l'algorithme préservant l'ordre des sommets terminaux varie comme N^2 pour une arborescence binaire [Knuth 73, p. 435]. D'autre part l'expression des contraintes sémantiques sous une forme utilisable dans (2.4) n'est pas évidente, à moins que ce ne soit précisément l'arborescence externe d'un arbre de capture préexistant qu'il s'agit d'améliorer par des voies qui vont être indiquées. De toute manière, dans l'état actuel du développement de l'arbre Diogène les données de base nécessaires (probabilités respectives des divers messages) font partiellement défaut. En outre, des remaniements apportés sans ménagement à des arbres conçus et mis au point parfois non sans peine à cause des interactions avec des chaînes d'application appelées en cours de descente, conduiraient à une instabilité plus nuisible qu'un certain manque d'efficacité à la saisie du message. Seules des interventions prudentes entrent donc en ligne de compte, inspirées en particulier des deux principes suivants (dérivés de conditions nécessaires d'optimalité) :

1. Permutations de sous-arborescences visant à faire migrer vers le haut de l'arborescence externe les sous-arborescences les plus utilisées; ceci revient à établir des raccourcis menant aux régions les plus visitées de l'arbre.
2. Applications à des sous-arborescences particulières des méthodes d'optimisation (avec ou sans contraintes) décrites précédemment. Le choix de ces sous-arborescences est "manuel".

Ces principes peuvent naturellement s'appliquer de manière combinée.

Exemple : L'arborescence "sémantique" de la figure 2-3 permet d'illustrer cette démarche de façon quasi caricaturale. Le premier principe conduirait par exemple à permuter la sous-arborescence "radio membre" avec une sous-arborescence "prescription" ou "admission", avec l'inconvénient que le choix "radio" doit être changé en "autres radios/prescriptions", lequel frise l'informulable. Mieux vaudrait peut-être réunir "prescription" et "admission", d'où l'arborescence de la figure 2-5. La seconde voie consiste à rechercher une optimisation interne à la sous-arborescence "radio" du genre de celle que présente la figure 2-6.

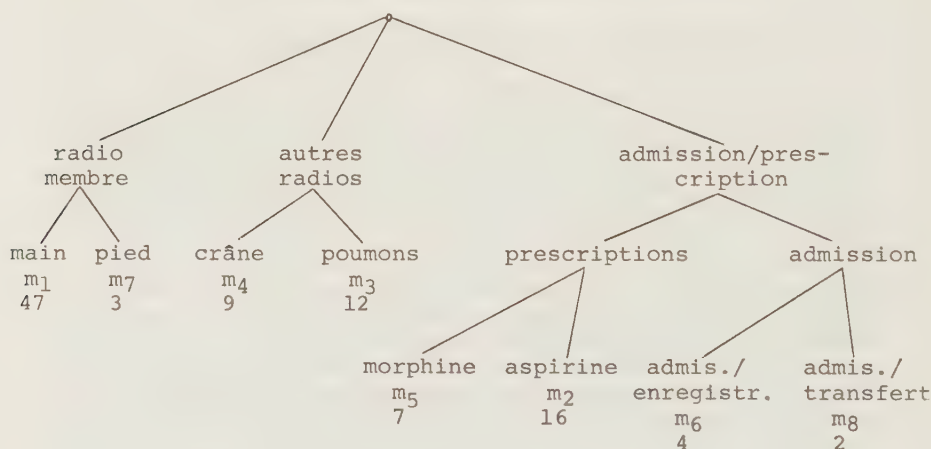


Fig. 2-5 Arborescence sémantique de la figure 2-3 améliorée par permutation de sous-arborescences. Base $a=3$. Longueur de cheminement : 2,29.

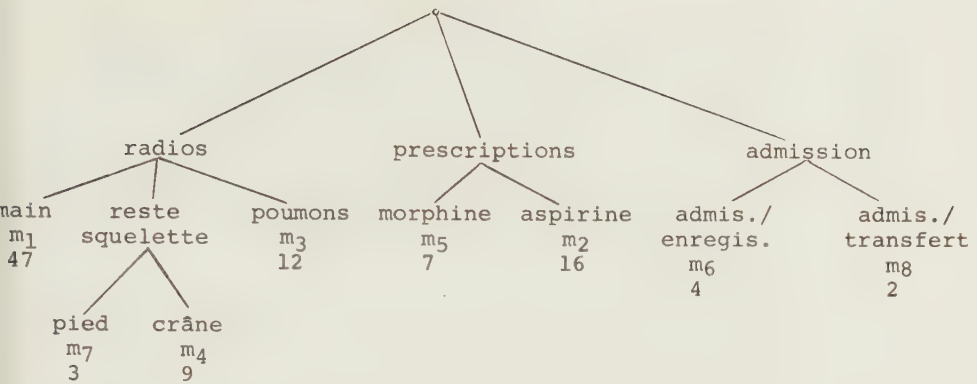


Fig. 2-6 Arborescence sémantique de la figure 2-3 améliorée par remaniement interne de la sous-arborescence "radios". Base $a=3$. Longueur de cheminement : 2,12.

2.2.4 Notations utilisées dans le paragraphe 2.2

- $\mathcal{A}$ alphabet, par exemple celui des caractères étiquetant les choix offerts sur les pages type-S
- a cardinal de $\mathcal{A}$ ou base d'arborescence externe (noté "a" hors des formules pour le distinguer de la forme verbale homonyme)
- $a,b,c..$ caractères de l'alphabet $\mathcal{A}$
- c_j caractère de l'alphabet $\mathcal{A}$ ($j=1...a$)
- E ensemble des sommets terminaux ou réponses d'un questionnaire
- E_i sous-ensemble de $\mathcal{M}$ (donc de E , par analogie) : classe de messages définie par des considérations sémantiques
- E_{ju} une des a sous-classes de E_i suivant une partition (définie par la sémantique) de E_i en a sous-ensembles
- F ensemble des sommets intérieurs ou questions d'un questionnaire
- I quantité d'information

$I(Q)$	information transmise par le questionnaire Q
$J(Q)$	information traitée par le questionnaire Q
$K(Q)$	information absorbée par le questionnaire Q
ℓ	taille moyenne des chaînes des messages de $\mathcal{M}$ codés dans l'alphabet $\mathcal{A}$; ou longueur de cheminement
ℓ_i	taille de la chaîne r_i représentative de m_i dans l'alphabet
ℓ_{ij}	longueur de cheminement de l'arborescence optimale couvrant $\mathcal{M}_{ij}$
$\ell_{i_u j_u}$	longueur de cheminement de l'une des $\underline{a}$ sous-arborescences optimales dont les racines sont les successeurs de X_{ij}
$\ell_{\min}$	taille moyenne des chaînes représentant les messages de $\mathcal{M}$ suivant un codage optimum dans l'alphabet $\mathcal{A}$; longueur de cheminement dans l'arborescence optimum couvrant $\mathcal{M}$
$\mathcal{M}$	ensemble des messages acceptables par le système à un moment donné
$\mathcal{M}_{ij}$	sous-ensemble $\{m_i \dots m_j\}$ de l'ensemble $\mathcal{M}$ des messages, où $\mathcal{M}$ est supposé ordonné
m_i	un message $\in \mathcal{M}$
N	cardinal de $\mathcal{M}$
n	nombre moyen de sélections nécessaires pour générer un message pris au hasard dans $\mathcal{M}$
n_i	nombre de sélections nécessaires pour générer m_i
p_i	probabilité du message m_i
P_Γ	valuation de questionnaire, probabilité définie sur l'ensemble Γ des arcs
Q	questionnaire
r_i	chaîne représentative de m_i dans l'alphabet $\mathcal{A}$
X	ensemble des sommets d'un graphe
X_{ij}	racine de l'arborescence optimale couvrant $\mathcal{M}_{ij}$

(X, Γ)	graphe
(X, Γ, P_Γ)	questionnaire
Γ	ensemble des arcs d'un graphe
Γ_i^{-1}	ensemble des arcs entrant au sommet i
Γ_i	ensemble des arcs sortant au sommet i

2.3. Graphe de capture et production du message

Une présentation formelle du mécanisme de descente et de production du message (MBG) est doublement profitable :

- a) elle conduit à un graphe (graphe de capture) représentatif de l'AC, rendant compte des aspects essentiels de l'enchaînement des pages au cours de la descente, alors que les apparences seules interviennent dans l'arborescence externe dont il a été question aux paragraphes précédents;
- b) une telle présentation fait ressortir la relation existant entre la structure de l'AC d'une part, et celle du message en tant que MBG suivant la syntaxe ACAP, d'autre part.

Certains points de ce paragraphe ont été abordés dans l'article [Berney/Baud/Scherrer 77]; ils sont traités ici de manière plus approfondie. Le point de vue choisi pour cette présentation, est celui de la constatation des faits (ou tout au moins de ceux d'entre eux jugés pertinents) observables dans la chaîne de tâches EXAC dont le code, au départ, a été conçu de manière relativement empirique.

2.3.1. Sommaire de la syntaxe ACAP

La syntaxe ACAP est justifiée intuitivement au paragraphe 1.4.7; telle qu'elle est présentée ici, elle s'écarte un peu de la syntaxe utilisée pratiquement dans le projet Diogène; en particulier certains éléments secondaires sont omis dans cette description. Ces divergences mineures permettent de faire ressortir plus commodément la relation entre MBG et AC.

En notations BNF, la syntaxe ACAP est décrite par les règles de production suivantes :

```

<message MBG> ::= (<ensemble de sections>)
<ensemble de sections> ::= <section> | <ensemble de sections>,
                                <section>
<section> ::= &<étiquette>=<complément> | <section>, <complément>
<complément> ::= <maillon> | (<ensemble de sections>)
<maillon> et <étiquette> ::= <chaîne de caractères
                                alphanumériques>
  
```

Un "ensemble de sections" entre parenthèses est aussi appelé "corps de section composé".

Toutes les règles de production de la syntaxe ACAP n'ont pour membre de gauche qu'un symbole non terminal; il s'agit donc d'une grammaire non contextuelle (voir par exemple [Hughes 72] ou [Gross/Lentin 67]). La relation existant entre ce type de grammaire et les automates à pile de mémoire, laisse pressentir le rôle qu'EXAC, avec la pile auxiliaire des pages, peut jouer dans la production d'un message suivant la syntaxe ACAP.

Du strict point de vue d'EXAC, en première approximation, le MBG est formé par concaténation des "info - BG" rencontrées dans les sélections successives d'une descente dans l'arbre :

$$\langle \text{MBG partiel} \rangle ::= \langle \text{MBG partiel} \rangle \langle \text{info-BG} \rangle .$$

Le contenu de ces contributions "info-BG" est déterminé par les instructions correspondantes au niveau source DESPAC, donc par les rédacteurs de pages. Toutefois la concaténation réalisée par EXAC n'est pas simplement passive, puisqu'il met en place les virgules séparant les sections ainsi que les parenthèses droites bornant les corps de section composés :

$$\langle \text{MBG partiel} \rangle ::= \langle \text{MBG partiel} \rangle \langle \text{apport EXAC} \rangle \langle \text{info-BG} \rangle$$

$$\langle \text{apport EXAC} \rangle ::= \emptyset \mid , \mid) \langle \text{apport EXAC} \rangle .$$

La dernière relation exprime que l'apport d'EXAC dans le processus de concaténation est soit vide, soit une virgule éventuellement précédée d'un certain nombre de parenthèses droites. Il est vide si le MBG partiel se termine par une étiquette ou une parenthèse gauche. En principe ces conditions sont détectables par examen de la chaîne formant une contribution; en pratique, dans l'instruction INFO-BG de DESPAC, le rédacteur de page établit explicitement la distinction entre les maillons et les autres types de contributions (étiquettes suivies ou non de parenthèses gauches).

Grâce à la fermeture automatique, par EXAC, des parenthèses droites du message MBG, la structure parenthétique de ce message reflète la structure de l'arbre de capture; c'est ce point qu'il s'agit de préciser.

2.3.2 L'automate de capture

Une manière épurée de présenter l'algorithme utilisé en cours de descente (dont le détail est donné au paragraphe 4.3.2) consiste à assimiler EXAC à un automate à pile de mémoire particulier : l'automate de capture. Cette transcription préserve essentiellement le mécanisme d'enchaînement des pages et celui de production du MBG.

Composants de l'automate

- a) Une bande d'entrée contenant la suite des choix faits par l'opérateur pendant la descente; chaque cellule est un caractère étiquetant une sélection; cette bande, étrangère à l'appareillage d'un automate à pile de mémoire ordinaire, matérialise la "liberté de choix" [Hughes 72, p. 27] nécessaire à tout automate fonctionnant en production. Cette liberté de choix est, bien entendu, en réalité le libre arbitre du donneur d'ordre.
- b) Une bande de sortie où s'écrit le message sous forme MBG; chaque cellule contient un caractère de la chaîne du message produit;
- c) Une bande mémoire (ou pile) S de pages successeurs dont les cellules reçoivent des images de pages d'AC.
- d) Une bande mémoire (ou pile) M pour la construction du message; ses cellules reçoivent les caractères spéciaux (virgules et parenthèses droites) qu'EXAC adjoint au MBG.
- e) Un organe E d'examen de la page successeur "courante", réalisant le détail des opérations qu'il serait trop fastidieux de décrire en termes de déplacements de bandes à gauche ou à droite.

Les bandes mémoires S et M pourraient être fondues en une dont les cellules seraient celles de S et M intercalées; l'automate de capture se ramène bien à un automate à pile de mémoire; l'usage de deux bandes mémoires simplifie le fonctionnement et met en évidence les deux éléments distincts mais coordonnés que sont l'enchaînement des pages et la production du message structuré.

Les bandes d'entrée et de sortie ne se déplacent, par convention, que vers la gauche, et ne sont sujettes qu'à lecture et écriture respectivement. Les bandes mémoires se déplacent à droite pour l'écriture et à gauche pour la lecture; la dernière cellule écrite sur une bande mémoire est aussi la première lue (cellule "courante").

Etat initial

- a) Les caractères étiquetant les choix successifs sont préécrits sur la bande d'entrée; la plus à gauche des cellules remplies est devant la tête de lecture.
- b) Une parenthèse gauche, premier caractère du MBG, est écrite sur la bande de sortie.

- c) La bande mémoire S contient une cellule "vide" dont le rôle apparaîtra plus loin, et, à gauche de celle-ci, la page successeur courante initiale, prête à l'examen par E.
- d) La tête de lecture/écriture de la bande mémoire M scrute une cellule contenant une parenthèse droite; la cellule de droite est "vide".

Fonctionnement

- a) Examen par E de la page (ou cellule) courante sur la bande mémoire S; s'il s'agit de la cellule vide s'arrêter; sinon, aller en b).
- b) Lecture d'une cellule sur la bande d'entrée; soit a_j le caractère qu'elle contient.
- c) Examen par E de la sous-cellule ou sélection a_j de la page successeur courante. Cette sélection contient :
 - c1. une chaîne de caractères "info-BG"; elle est copiée sur la bande de sortie; cette chaîne peut ouvrir $P \geq 0$ parenthèses gauches (en général P est 0 ou 1); P parenthèses droites sont écrites sur la bande mémoire M;
 - c2. la spécification de $N \geq 0$ pages successeurs; ces N successeurs sont écrits sur la bande mémoire S; N virgules sont écrites sur la bande mémoire M.
- d) Copie de la bande mémoire M sur la bande de sortie, jusqu'à, et y-compris, la première virgule ou cellule vide de M (c'est la réalisation de l'apport EXAC).
Retour en a).

Conditions d'arrêt

Selon le mode de fonctionnement indiqué, l'automate s'arrête à la lecture de la cellule vide sur la bande mémoire S. Ceci suppose que la bande d'entrée est assez longue. Cette hypothèse n'a rien de restrictif eu égard à la signification concrète de son contenu : le donneur d'ordre fait un choix sur chacun des affichages qui lui sont présentés; son libre arbitre se manifeste par ce choix, non par la décision d'arrêter ou de suspendre la descente. D'ailleurs l'automate ignore la durée : une suspension momentanée de descente n'implique pas l'arrêt de l'automate.

Note

Tel qu'il est décrit l'automate de capture autorise une virgule dans le MBG quel que soit le dernier caractère du MBG partiel; l'algorithme réel doit naturellement éviter qu'une virgule suive une étiquette ou une parenthèse gauche afin de respecter la syntaxe ACAP.

2.3.3 Graphes déduits du fonctionnement de l'automate

2.3.3.1 Enchaînement des pages et graphe de capture

Abstraction faite de la production du message, le fonctionnement de l'automate traduit l'exploration d'un graphe biparti (annexe B), le graphe de capture, construit selon les conventions suivantes :

- a) Une page, contenu d'une cellule de la bande mémoire S, est représentée par un sommet de la classe P du graphe (cercles de la figure 2-7).
- b) Les informations d'une sous-cellule ou sélection dans une page sont représentées par un sommet de classe C (rectangles de la figure 2-7).
- c) L'ensemble formé d'un sommet de classe P et de ses sommets successeurs (nécessairement de classe C) représente une page avec les choix qu'elle offre.
- d) L'ensemble formé d'un sommet de classe C et de ses sommets successeurs (de classe P) représente une sélection avec les pages qu'elle désigne comme successeurs. Par convention, l'ordre des sommets successeurs, de gauche à droite, est celui des pages sur la bande mémoire S.

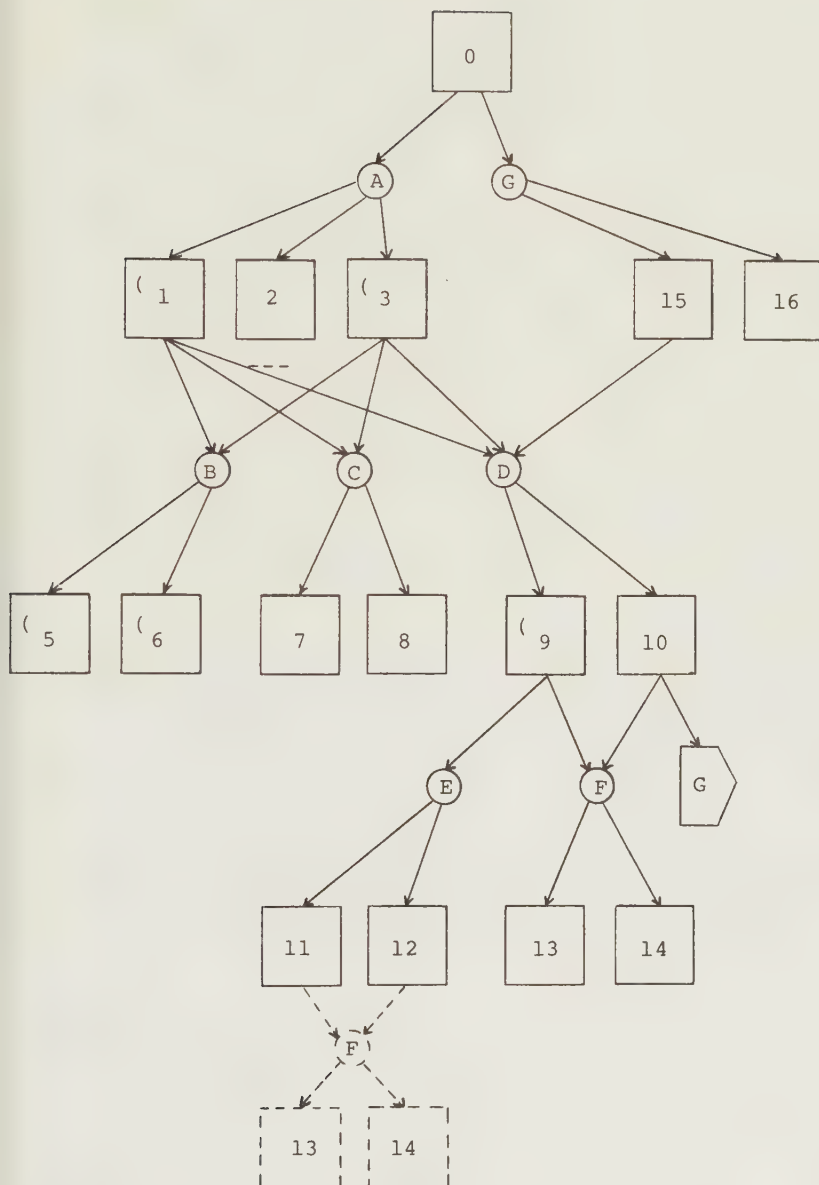


Fig. 2-7 Graphe de capture. Les pages sont désignées par des lettres A, B, ..., les choix par des nombres 1, 2, ... La page F est représentée dans deux variantes conduisant à la même arborescence externe (figure 2-8). Une parenthèse gauche dans le carré représentatif d'une sélection annonce l'ouverture d'une parenthèse ACAP dans la contribution au MBG (info-BG) de cette sélection.

2.3.3.2 Arborescence externe

Par définition, l'arborescence externe introduite au paragraphe 1.3, est l'arborescence des chemins dans le graphe de capture exploré suivant l'algorithme décrivant le fonctionnement de l'automate de capture.

La figure 2-8 représente l'arborescence externe déduite du graphe de la figure 2-7. Chaque page (sommet circulaire) du graphe de capture admet autant de sommets images dans l'arborescence externe qu'il existe de chemins dans le graphe de la figure 2-7 passant par cette page; ainsi B dans 2-7 donne B et B₁ dans 2-8 à cause des deux arcs incidents à B dans 2-7. Tout arc issu d'un sommet "page" de 2-7 (avec le sommet "sélection" attaché à cet arc) représente un choix possible (libre arbitre) de la part de l'opérateur, donc le départ d'un chemin possible dans l'arborescence externe. Au contraire, tous les arcs issus d'un sommet "sélection" de 2-7 (et les sommets pages qui leur sont attachés) alimentent la pile ou bande mémoire S; ainsi, B, C et D, au même niveau dans 2-7, ont pour images dans 2-8 des sommets descendant les uns des autres; d'où la prolifération des images de C (4 images), D (8 images) etc. dans l'arborescence externe, dont le volume rend toute représentation malaisée.

L'arborescence externe ne rend pas compte du mode d'enchaînement des pages, parce que la correspondance entre elle et le graphe de capture n'est pas bijective; les deux variantes du graphe de la figure 2-7, par exemple, produisent la même arborescence externe.

2.3.3.3 Description structurelle du message

Désignons par "label_i(inf_i" ou "inf_i" la contribution au MBG de la sélection i d'un graphe de capture tel que celui de la figure 2-7, selon que cette contribution ouvre ou n'ouvre pas de parenthèse; "label_i" est de la forme &<étiquette>=, alors que "inf_i" est de la forme &<étiquette>=<maillon>.

En admettant que la suite des sélections faites par l'opérateur sur les pages de l'arbre de capture correspondant au graphe 2-7 soit : 3, 6, 7, 9, 12, 13, 15, 9, 12, 13, 16, ..., le message produit sur la bande de sortie par l'automate de capture a la forme suivante

```
(2.5) ...label3(inf3,label6(inf6),inf7,label9(inf9,inf12,
      inf13)),inf15,label9(inf9,inf12,inf13),inf16,...
```

Ce message présente une structure de "Liste" au sens de [Knuth 68, pp. 312 et 406] dont la figure 2-9 donne le diagramme; ce diagramme est aussi la "description structurelle" du message selon l'expression de [Moreau 75, p. 90].

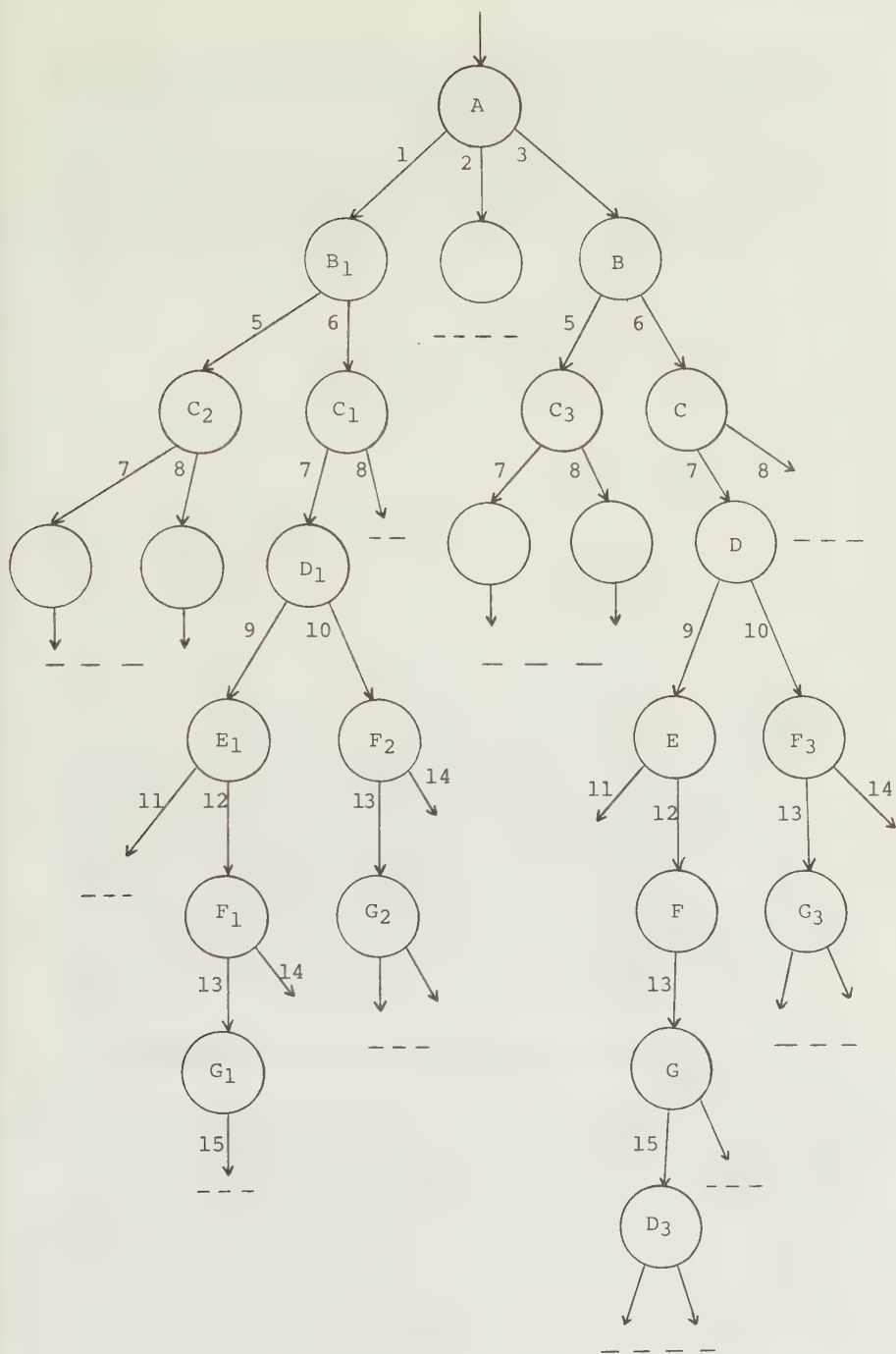


Fig. 2-8 Arborescence externe déduite du graphe de la figure 2-7.

Quelque nom qu'il prenne ce schéma représente la structure du message considéré comme donné a priori, c'est-à-dire sans égard au mode de production.

Evidemment le lien relevé ici entre structures de "Listes" d'une part, et grammaires non contextuelles (donc automates à pile de mémoire) d'autre part, ne s'applique pas qu'au MBG, mais dans tous les cas où ces structures apparaissent.

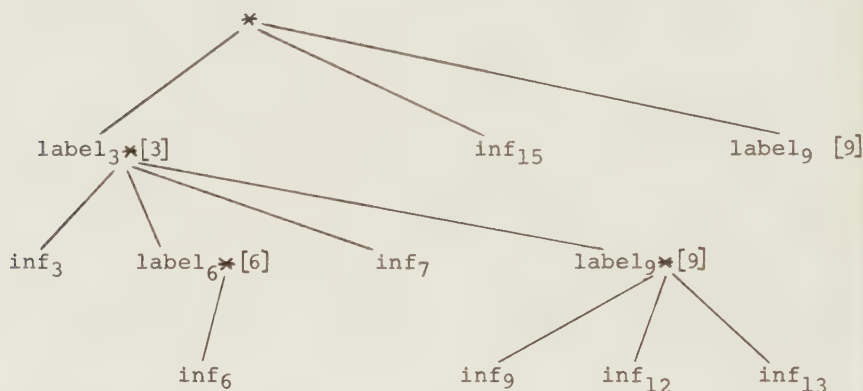


Fig. 2-9 "Diagramme de Liste" ou "description structurée" du message produit par la suite des choix 3,6,7,9,12,13,15,9,12,13 dans l'arbre de capture correspondant à 2-7. Les * annoncent la définition d'une Liste, désignée par un nombre entre crochets et opposée à une atome tel que inf₇. Dans cet exemple il y a deux références à la Liste [9] ; l'une d'elle s'accompagne de la définition de cette Liste. Il y a une seule référence aux autres Listes [3] et [6] .

2.3.4 Graphe de capture produisant une classe donnée de messages

Partant d'un graphe de capture et de l'automate qui l'exploire on a fait ressortir la structure du message qu'il produit. Du point de vue de la conception d'un AC, le problème inverse est plus important. Il s'agit alors, partant d'un message de forme donnée, de trouver le graphe de capture susceptible de générer ce message. Cette "forme donnée d'un message" définit une classe de messages, c'est-à-dire un ensemble de messages ayant même structure parenthétique et mêmes étiquettes. Les messages d'une classe diffèrent donc par le contenu de leurs maillons

(au sens de la définition du paragraphe 2.3.1) et, accessoirement, par l'ordre des sections dans un corps de section composé. On pourrait également englober, dans une classe, des messages présentant un nombre variable d'occurrences d'une section d'étiquette donnée, mais cette extension n'apporte rien de fondamental au concept.

Le message (2.5) appartient ainsi à la classe dont la forme, pour un ordre particulier des sections constitutives, est la suivante :

$$(2.6) \quad \dots \text{label}_3 \left(\begin{Bmatrix} \text{inf}_1 \\ \text{inf}_3 \end{Bmatrix} , \text{label}_6 \left(\begin{Bmatrix} \text{inf}_5 \\ \text{inf}_6 \end{Bmatrix} \right) , \begin{Bmatrix} \text{inf}_7 \\ \text{inf}_8 \end{Bmatrix} , \right. \\ \left. \text{label}_9(\text{inf}_9, \begin{Bmatrix} \text{inf}_{11} \\ \text{inf}_{12} \end{Bmatrix} , \begin{Bmatrix} \text{inf}_{13} \\ \text{inf}_{14} \end{Bmatrix}) , \text{inf}_{15} , \right. \\ \left. \text{label}_9(\text{inf}_9, \dots) , \dots \right.$$

Il est supposé que inf_1 et inf_3 , par exemple, ont même étiquette, autrement dit que

$$\begin{Bmatrix} \text{inf}_1 \\ \text{inf}_3 \end{Bmatrix} \quad \text{signifie} \quad \<\text{étiquette}_{1,3}\> = \begin{Bmatrix} \<\text{maillon}_1\> \\ \<\text{maillon}_3\> \end{Bmatrix} .$$

Concrètement une classe de messages pourrait être la prescription d'un médicament pour un patient, précisant une durée de traitement et une dose par prise; tous les messages de cette classe ne diffèrent que par le nom du médicament, le nom du patient, la dose appliquée etc., la structure parenthétique et la nature des informations (donc les étiquettes) restant les mêmes. En terminologie linguistique on peut dire que les accolades, dans (2.6), esquissent l'axe paradigmatique du message.

Le problème posé est donc de remonter de la description structurale (figure 2-9) que la forme donnée de la classe permet de déduire immédiatement, au graphe de capture (figure 2-7). Le passage peut s'opérer en deux phases.

Première phase

Dans cette première phase on construit le graphe générateur d'un message de la classe, sans égard à l'axe paradigmatique. On peut admettre que, dans ce message unique, chaque maillon reçoit une valeur symbolisant toutes les valeurs possibles de ce maillon dans la classe considérée. Cette phase elle-même se déroule en deux temps :

- a) Construction du graphe partiel des définitions de Listes apparaissant dans la description structurale; chaque définition donne un sommet du graphe partiel, et les arcs traduisent la hiérarchie des Listes dans le diagramme, comme le montre l'exemple de la figure 2-10.

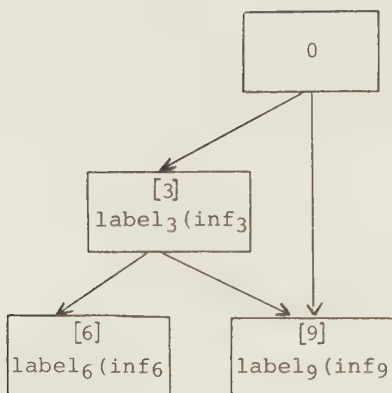


Fig. 2-10 Graphe partiel des Listes, dérivé de la figure 2-9. Chaque définition de Liste dans la figure 2-9 donne un sommet du graphe partiel et contribue au message par son label et son premier atome.

b) Adjonction des atomes. Le premier atome de chaque Liste peut être fusionné, dans le graphe partiel, avec le sommet associé à cette Liste (figure 2-10). Les autres atomes forment de nouveaux sommets placés, au choix, selon l'une ou l'autre des règles suivantes :

- création d'un nouveau sommet terminal, successeur du sommet représentant la Liste où figure l'atome;
- création d'un sommet non terminal, intercalé entre le sommet représentant la Liste et un sommet représentant un autre atome (par la règle précédente) ou une Liste imbriquée dans celle où figure l'atome considéré.

Les figures 2-11 à 2-13 donnent trois exemples de graphes générateurs construits à partir de 2-10 par adjonction des atomes suivant les règles précédentes. Dans les deux premières l'ordre des sections dans (2.5) et (2.6) est préservé.

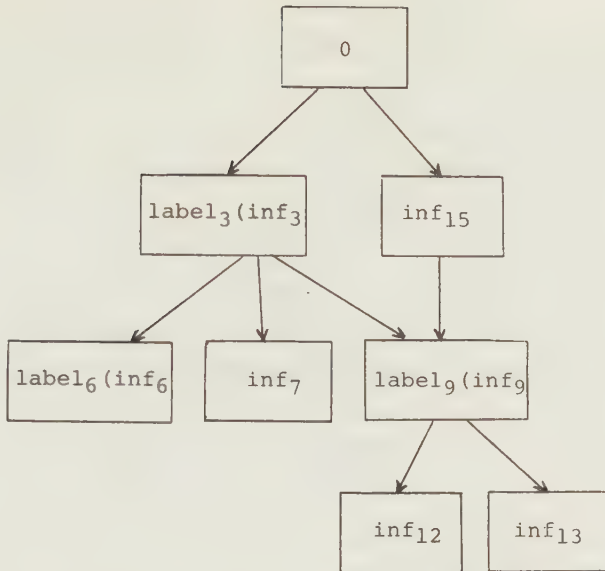


Fig. 2-11 Graphe générateur du message (2.6) et conduisant au graphe de capture de la figure 2-7.

Seconde phase

En comparant les figures 2-11 et 2-7 on constate que le graphe générateur est un graphe partiel du graphe de capture dans lequel ne subsiste qu'un sommet sélection de chacune des pages. Le graphe de capture pour une classe de messages se déduit donc du graphe générateur en intercalant sur chaque arc de ce graphe un sommet page, et en accrochant à chacun de ces nouveaux sommets autant de nouveaux sommets sélections que l'exige l'ensemble des valeurs possibles sur l'axe paradigmatique au point correspondant du message.

A titre d'exemple, la figure 2-14 montre le graphe de capture déduit du graphe générateur (dans sa variante de la figure 2-11) pour la classe de messages (2.6). On retrouve le graphe de la figure 2-7, à l'exclusion, naturellement, des sommets qui, dans ce dernier graphe, conduisaient à des messages sortant de la classe considérée.

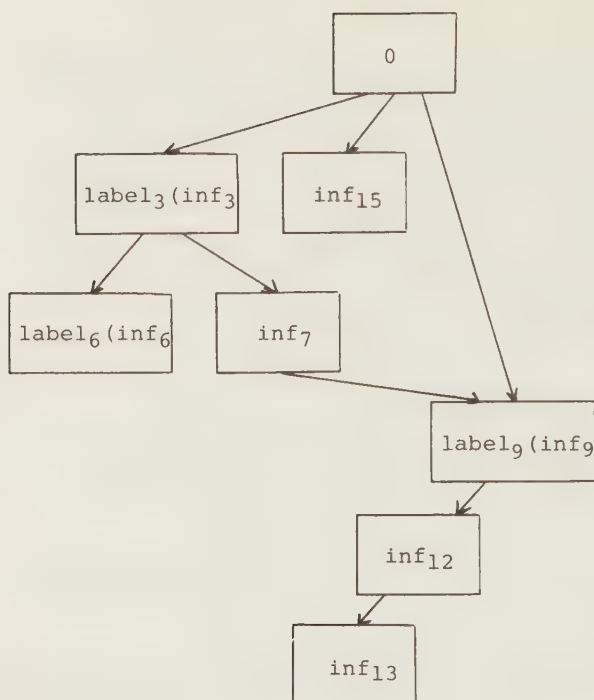


Fig. 2-12 Variante du graphe générateur du message (2.6).

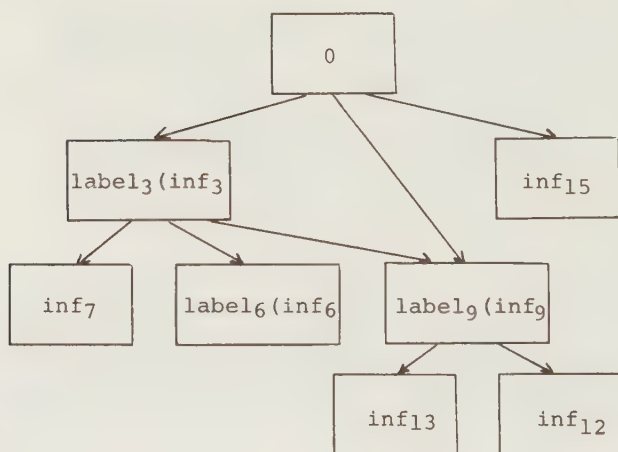


Fig. 2-13 Variante du graphe générateur du message (2.6) ne préservant pas l'ordre des sections. L'adjonction des atomes a été faite systématiquement par création d'un sommet terminal adjacent au sommet représentant la liste propriétaire de l'atome.

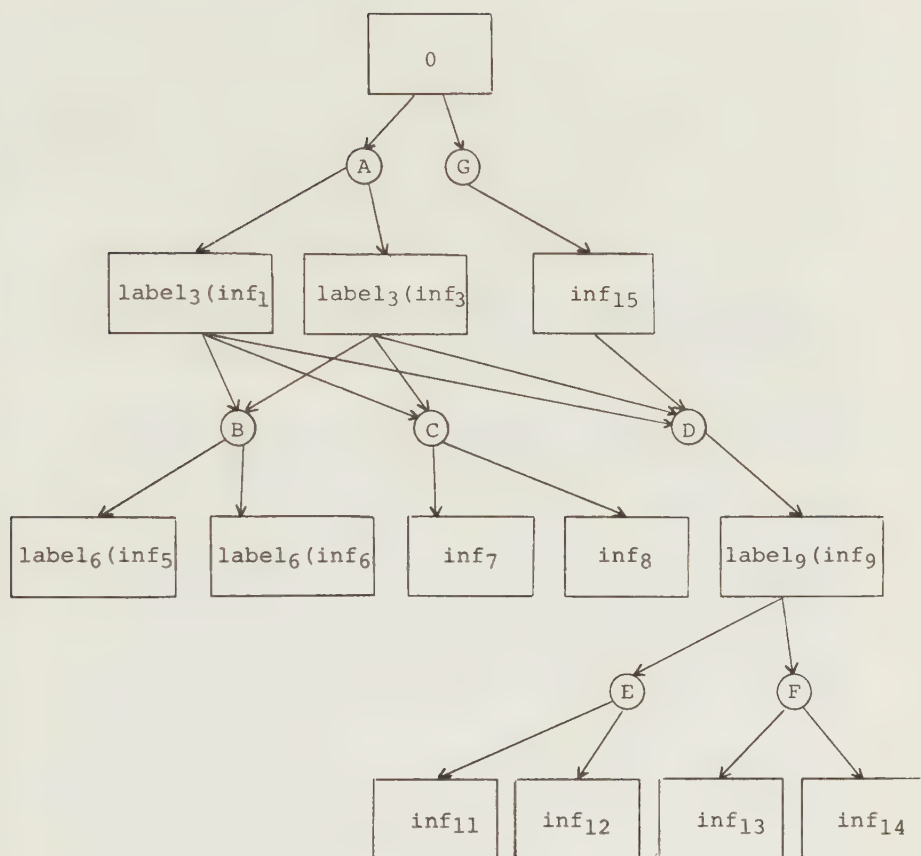


Fig. 2-14 Graphe de capture déduit du graphe générateur de la figure 2-11, pour la classe de messages (2.6).

Conclusion

Le graphe de capture produisant une classe donnée de messages n'est en général pas unique, même si l'on impose un ordre aux sections dans le message. Si, parmi tous les arbres de capture générant la classe considérée de messages, on admet que le meilleur est celui dont le plus grand nombre de pages possible ne désignent aucun successeur de manière à en assurer la portabilité, on dispose d'un critère permettant de ne conserver (et de ne construire) qu'un graphe, pour un ordre particulier des sections du message.

De ce point de vue, le graphe générateur de la figure 2-13 est préférable à ceux des figures 2-11 et 2-12 : il conduit à un arbre de capture avec 5 pages sans successeur au lieu de 4 dans 2-11; et de 3 dans 2-12.

Le "meilleur" graphe s'obtient, lors de l'adjonction des sommets représentant les atomes dans le graphe générateur, en les plaçant tous comme successeurs du sommet de la Liste où ils figurent.

3. Constitution de librairies de pages

3.1. Rôle de la librairie de pages dans le système Diogène

Dans la conception d'un système transactionnel, l'emploi de l'arbre de capture a pour effet de marquer de façon précise les limites entre programmes d'application d'une part, et interface avec le monde extérieur d'autre part. Dans un système ordinaire, cette interface (les programmes qui l'anime comme le contenu du dialogue) se trouve être diffuse dans les programmes d'application. Dans un système comme Diogène, ils forment au contraire des entités bien délimitées : les programmes d'interface constituent la chaîne EXAC, alors que le contenu du dialogue, au moins dans la direction "système-donneur d'ordre", est matérialisé par la librairie de pages.

L'existence propre d'EXAC et de la librairie des pages a des conséquences importantes dans l'analyse et la mise sur pied des applications du projet Diogène :

- a) A condition de remplacer provisoirement par des pages fixes ce qui deviendra peut-être des pages dynamiques ou squelettes (paragraphe 1.4.4), il est possible de présenter au futur utilisateur la prise des messages de transaction, dans un domaine particulier, avant tout développement important de programmes d'application ou de structures de la banque de données; ceci ne signifie pas que l'élaboration de ces pages soit un travail négligeable;
- b) Pour autant qu'ils interviennent dans la prise des messages, les programmes d'application sont des auxiliaires de remplissage des pages ou de validation partielle du message;
- c) Une fois le message de transaction constitué, syntaxe ACAP (paragraphe 1.4.7), le travail de la chaîne de validation sémantique est simplifié, et la structure de cette chaîne, fortement marquée par celle du message.

En d'autres termes, la démarche suivie dans l'analyse et la mise sur pied d'une fonction donnée va de l'extérieur vers l'intérieur et permet d'y associer assez tôt des utilisateurs non informaticiens. En l'absence d'arbre de capture on s'efforcera sans doute de prendre l'avis de ceux que le système va toucher en leur soumettant des spécifications externes, au risque toutefois qu'il leur soit difficile de se prononcer en l'absence de démonstration concrète. D'autre part, l'analyse va progresser de l'intérieur vers l'extérieur, c'est-à-dire commencer par le programme d'application qui, à tel stade d'exécution éprouve le besoin de telle information, et envoie tel affichage pour la recueillir.

Définies ainsi pour elles-mêmes, maintes pages sont utilisables indépendamment du "contexte" de l'application où elles interviennent. Ce souci de portabilité des pages, déjà mentionné au paragraphe 1.4.6 ramène au troisième aspect de l'économie de la prise de messages évoqué au paragraphe 2.1 : l'effort de développement de l'interface donneur d'ordre/système transactionnel. Sans contester le caractère subjectif et qualitatif de cette appréciation, nous pouvons conclure des considérations précédentes, que l'approche "arbre de capture" est économique en fait d'analyse et de développement.

Ce chapitre montre comment, dans le projet Diogène, peuvent être constituées des librairies de pages propres à des applications conçues de manière relativement indépendantes mais utilisant des ensembles de pages communs, librairies qui sont ensuite fondues en une pour le système transactionnel global.

3.2. Structure des librairies de pages

Au paragraphe 1.4.5 ont été indiquées les conditions auxquelles doit satisfaire le mécanisme de localisation des pages en librairie. Ces contraintes dictent de manière relativement précise, la structure qu'il convient de donner aux librairies de pages.

1. Une contrainte de base est la nécessité, en cours de descente, de pouvoir lire toute page de la librairie vers la mémoire centrale, par un seul accès physique en mémoire de masse; il en résulte, si l'on veut éviter de garder en mémoire centrale un index volumineux, que les pages de la librairie doivent être des articles de taille fixe; une page est localisable dès lors que l'on connaît son numéro séquentiel dans la librairie.
2. Au niveau source DESPAC, il faut que le rédacteur de pages puisse désigner les pages de manière parlante, par des noms mnémoniques, ceci soit dans les instructions DESPAC déclarant une page donnée, soit dans les références faites de cette page à d'autres pages (successeurs).

Le problème est donc maintenant de trouver un mécanisme pour passer du nom mnémonique d'une page à un numéro séquentiel en librairie en tenant aussi compte des aspects subsidiaires suivants :

- il faut éviter le gaspillage de la mémoire de masse;
- l'indexation d'une désignation de page (voir paragraphe 1.4.5, alinéa c4) doit être possible.

D'où le mécanisme retenu :

- a. Le fichier librairie est partitionné. Une partition, connue sous le nom de "groupe" de pages, comprend en général l'ensemble des pages intéressant un domaine d'application spécifique ou des ensembles de pages à usage multiples (pages éminemment portables) : prise de la date, spécification d'une unité de soins, etc.
- b. Lors de la création ou de la restructuration d'une librairie de pages, l'utilitaire CHAPAC (paragraphe 1.4.5, alinéa b) accepte comme directives le nombre de pages prévu dans chacun des groupes. A ce niveau, les groupes sont identifiés par des codes numériques. Sur ces données, CHAPAC forme la "table des adresses de groupes", TAGR, qui n'est autre que la suite, par code de groupe, des numéros séquentiels en librairie de la première page de chacun des groupes. Cette table, placée au début de la librairie, est lue par EXAC pour toute descente et gardée en mémoire centrale. Le nombre total de groupes étant réduit, la table a une taille acceptable (actuellement 50 groupes admis, soit, à raison de 30 bits par entrée, 25 mots de 60 bits pour la table). La table TAGR condense la structure de la librairie : les groupes s'y succèdent par ordre croissant de leur code. Elle permet donc, connaissant le code de groupe d'une page et son numéro au sein du groupe, de déterminer le numéro séquentiel absolu de la page. Le couple formé d'un code de groupe et du numéro relatif dans le groupe est l'identificateur compacté de la page.
- c. La table ne peut servir pendant la descente que si, dans les pages compilées, toutes les désignations de pages sont des identificateurs compactés. L'un des rôles du compilateur DESPAC est de convertir les désignations mnémoniques source des pages en identificateurs compactés. Les données nécessaires à cette conversion sont fournies à DESPAC par deux types d'instructions "globales", c'est-à-dire portant sur toute une compilation, alors que pour le reste, DESPAC se réinitialise au début de la compilation de chaque page, comme FORTRAN compile des sous-programmes indépendamment les uns des autres. Ces deux types d'instructions globales permettent notamment
 - de donner un nom symbolique aux groupes,
 - d'assigner une mnémonique à toute page repérée par le nom de son groupe et son numéro relatif dans le groupe.

Sans entrer dans des détails de syntaxe et de sémantique donnons un exemple de l'allure des premières instructions d'un fichier source DESPAC :

```
*  AFFECTATION DE NOMS AUX CODES NUMERIQUES DES GROUPES
GROUPES
```

```
LAB=4
```

```
PHA=10
```

```
UNI=3
```

```
*  AFFECTATION DE MNEMONIQUES AUX DESIGNATIONS DE PAGES
MNEMONIQUES
```

```
REQLAB=LAB12
```

```
CONTLAB=LAB9
```

```
DATGEN=UNIO
```

```
:
:
```

d. Conséquences de la méthode adoptée :

- le code numérique d'un groupe peut être changé par recompilation des pages qui le compose et de celles qui s'y réfèrent après changement d'une seule instruction, puis rechargement de la librairie par CHAPAC;
- une page ou un ensemble de pages désignées par des expressions de la forme "mnémonique" + "nombre" (où "nombre" peut être en fait un symbole) peuvent être déplacées dans un groupe par recompilation, après changement d'une seule instruction globale mnémonique;
- la restructuration d'une librairie (changement de la taille de groupes, y compris adjonction ou suppression d'un groupe entier) se fait sans recompilation, par simple rechargement de la librairie via CHAPAC instruit des directives voulues.

Dans l'ensemble ce mécanisme s'est montré commode à l'usage puisqu'il dispense les rédacteurs de se préoccuper de l'emplacement des pages en librairie sans nuire à l'efficacité : les rédacteurs de pages gardent malgré tout le contrôle de l'emplacement des pages, ce qui évite le gaspillage de mémoire de masse; enfin la lecture de n'importe quelle page, par un seul accès physique, demeure garantie.

A propos de la compilation des pages par DESPAC notons encore que la taille fixe choisie (actuellement 300 mots environ, mots de 60 bits) pour les pages en librairie, ne permet pas de recevoir une page qui offrirait simultanément sous tous les rapports le maximum de complexité. Après encodage des informations de contrôle de la page, DESPAC les compacte en tenant compte

- de l'absence de certaines sélections;
- du fait que, dans les sélections existantes, les successeurs désignés sont variables en nombre et en qualité etc;
- si les gains réalisés par compaction des informations de contrôle ne suffisent pas, DESPAC récupère encore l'espace occupé par les lignes blanches de l'affichage.

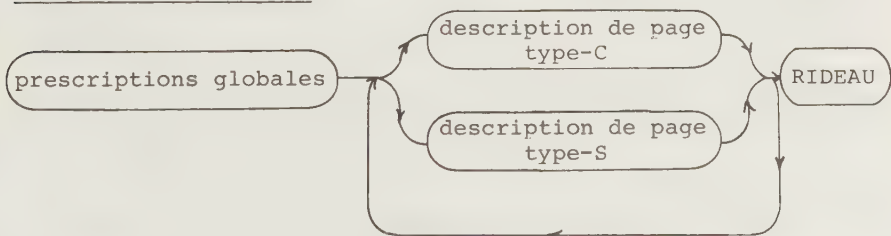
Finalement une page est représentée par une table de format éminemment variable.

3.3. Format source DESPAC

La description du format source DESPAC donnée dans ce paragraphe est superficielle et quelque peu approximative; les précisions et les détails n'ont d'intérêt que pour ceux qui sont amenés à rédiger des pages en vue des applications; les documents internes à la Division d'informatique donnent tous les renseignements voulus. Ce format source a été imaginé en l'absence de toute prétention à l'élégance comme un moyen relativement confortable, c'est-à-dire facile à apprendre et donnant des descriptions claires, de spécifier les pages d'un arbre de capture. [Le Beux 75] désigne par "langage de grille" un format source du même genre.

3.3.1 Aperçu de la syntaxe

fichier source DESPAC

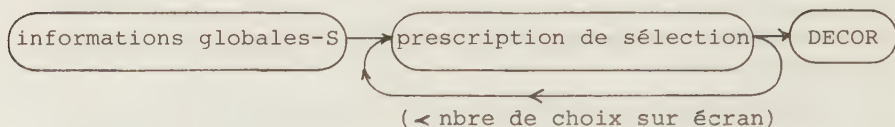
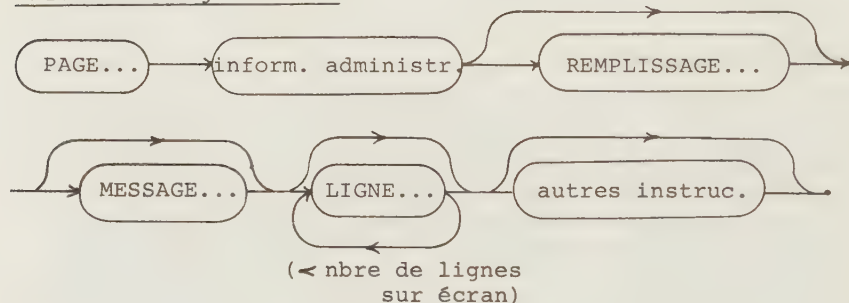
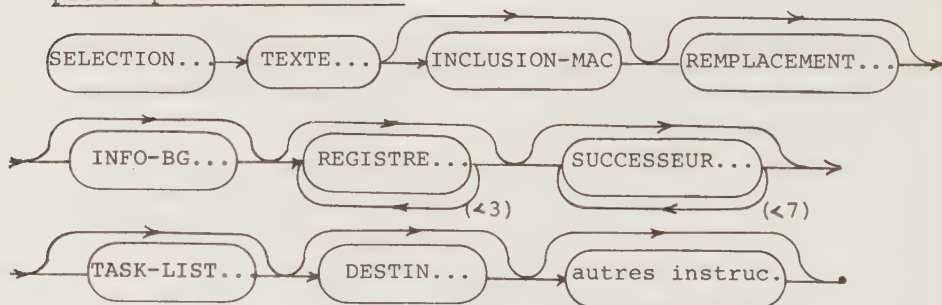


prescriptions globales

voir paragraphe 3.2.

description de page type-C

d'un intérêt trop limité dans la conception générale de l'arbre de capture pour être donnée ici

description de page type-Sinformations globales-Sprescription de sélection

3.3.2 Commentaires et aperçu sémantique

1. Un fichier source DESPAC commence par des prescriptions globales de groupes et mnémoniques (accessoirement aussi, de symboles) dont il a été question au paragraphe 3.2. Viennent ensuite les descriptions individuelles des pages; l'ordre et le nombre en est indifférent. L'instruction RIDEAU termine la compilation. On a déjà fait remarquer qu'il n'est pas nécessaire de compiler ensemble toutes les pages d'un arbre; l'utilitaire CHAPAC permet l'adjonction à une librairie existante, de pages nouvelles, compilées séparément.
2. La description source de toute page commence par l'instruction PAGE... et finit avec DECOR. DECOR provoque la compaction des tables générées préalablement en supposant que la page peut offrir le maximum de complexité et envoie la table ainsi compactée sur le fichier séquentiel des pages compilées, à condition qu'aucune erreur fatale ne soit apparue en compilant cette page. L'instruction

PAGE <désignation> TYPE-S [*<autres clauses>*]

identifie la page par une "désignation" en général de la forme "mnémonique" ou "mnémonique" + "nombre" ou encore "mnémonique" + "symbole"; elle en spécifie le type S ou C; d'autres clauses ou options annoncent par exemple une page "clef" (paragraphe 1.4.4.c).

3. Dans les informations globales doivent figurer des données administratives, gardées en librairie et reprises dans certaines analyses faites par l'utilitaire CHAPAC :

- nom du responsable de la page,
- date de création,
- titre caractérisant le contenu de la page.

L'instruction

REPLISSAGE <nom de chaîne> [*..*]

annonce une page squelette (paragraphe 1.4.4.b) dont les informations de remplissage sont recherchées par la chaîne d'application du nom donné.

MESSAGE <nbre de lignes> [*, <num. de ligne>*]

réserve à l'affichage du message de transaction en clair (MAC) le "nombre de lignes" demandé, soit au haut de l'écran, soit depuis le "numéro de ligne" donné.

Les prescriptions répétables

LIGNE <num. de ligne>,<contenu de ligne>

font apparaître le "contenu de ligne", en général un commentaire pour le donneur d'ordre, au "num. de ligne" indiqué. Un tel texte ne peut être sélectionné en cours de descente. Le compilateur DESPAC signale les conflits éventuels d'occupation de l'écran par MESSAGE, LIGNE et positions sensibles de l'écran.

4. Toute page type-S comporte une sélection au moins. Chaque prescription de sélection commence par

SELECTION <num. logique de sélection> [<options>]

Le "numéro logique de sélection" identifie la sélection indépendamment de la position physique sur l'écran du choix correspondant; on peut ainsi modifier l'allure de l'affichage, sans changer les références faites aux sélections par leurs numéros logiques, dans la définition d'un masque sur une page par exemple. Chaque numéro logique de sélection doit être unique dans une page donnée; l'ordre suivant lequel les sélections sont prescrites est indifférent.

Les positions physiques possibles des choix sur l'écran sont désignées par des numéros, de 1 au nombre maximum de ces choix (24 actuellement).
L'instruction

TEXTE <position de l'écran>,<contenu du texte>

indique la "position de l'écran" où doit apparaître le "contenu du texte" de la sélection. Cette position sur l'écran est un numéro absolu ou une expression dont la forme importe peu, mais dont l'intérêt est de la rendre relative aux angles de l'écran; on peut, par exemple, faire qu'une sélection occupe le coin inférieur droit de l'écran sans égard au nombre total de choix admissibles sur l'écran, ce qui est très utile si l'on passe d'un terminal de 16 à 24 lignes, et qu'on ne veut pas récrire les pages pour conserver aux choix sur l'écran la même position relative.

5. A part SELECTION et TEXTE tout est facultatif dans les prescriptions de sélection et ne touche qu'aux informations de contrôle de sélection, imperceptibles au terminal lors de l'affichage de la page, mais décisives quant aux actions prises par EXAC si la descente passe effectivement par cette sélection.

INCLUSION-MAC

provoque l'adjonction au message en clair MAC, du texte de la présente sélection, ou d'un texte de remplacement lorsque figure l'instruction

REPLACEMENT <texte de remplacement>

qui permet, si nécessaire, d'améliorer la forme du MAC.

INFO-BG <champs info-BG>

est le pendant de ce qui précède, mais pour le message de transaction passé aux chaînes d'application, le MBG (paragraphe 1.4.1 et 1.4.7); les "champs info-BG" définissent le contenu de l'adjonction au texte MBG (adjonction qui doit tenir compte des exigences de la syntaxe ACAP) et précisent, s'il y a lieu, le nombre de délimiteurs gauches ACAP ouverts dans cette contribution; en cours de descente cette dernière information est reprise par EXAC pour la fermeture automatique des délimiteurs droits correspondant aux délimiteurs gauches ouverts à un niveau donné du MBG.
L'instruction

REGISTRE <num. de registre><type de manipulation><opérande>

modifie le contenu du registre software (paragraphe 1.4.6) de numéro donné (0-7); ce contenu est une valeur d'indexation utilisée dans la désignation des pages successeurs, ou un masque; la manipulation consiste soit à entrer ou modifier une valeur d'indexation, soit à introduire ou altérer un masque dans le registre. Dans le premier cas l'opérande a la forme d'un nombre (ou de son symbole), dans le second, d'une suite de numéros logiques de sélections affectées par le masque. La nature des références non destructives faites aux registres software se précisera à propos de l'instruction suivante.

6. La syntaxe de l'instruction SUCCESSEUR s'est compliquée parallèlement aux efforts faits en cours de projet pour rendre les pages de plus en plus portables, c'est-à-dire indépendantes du contexte où elles interviennent; ceci implique, pour conserver un niveau donné de flexibilité, de pouvoir prescrire l'effet d'une sélection sur une page, dans le contexte où cette page est appelée, du moins dans une certaine mesure, contexte incarné principalement par l'instruction désignant la page en question comme successeur. Il est admis sept successeurs par sélection au maximum. En cours de descente, les désignations de pages successeurs sont placées par EXAC dans la pile auxiliaire (paragraphe 1.4.6) de telle sorte que

la première dans la description source se trouve au-dessus de la pile et soit ainsi le prochain affichage envoyé (sauf exceptions peu significatives); au passage d'une sélection sans successeur deux cas peuvent se présenter : soit la pile est vide, ou ne contient plus que des pages clefs en vue de l'enchaînement de transactions, et dans ce cas EXAC appelle la chaîne de validation sémantique/exécution spécifiée comme nous le verrons plus loin; soit la pile contient des désignations de pages résiduelles provenant de sélections antérieures ayant amenés plusieurs successeurs à la fois dans la pile; dans ce cas la descente se poursuit avec l'envoi de l'affichage prescrit par le sommet de la pile. L'instruction successeur a l'allure suivante

SUCCESEUR <désignation de page>

[INDEXE <registre soft.> {PILE
AFFICHAGE}]

[MASQUE {<masque explicite>
<registre soft.>}]

[REPETITION <nbre de répétitions>]

[FANTOME <position écran>] [<diverses autres clauses>]

La "désignation de page" est en général une mnémonique définie dans les prescriptions globales du fichier source DESPAC; la page ainsi visée n'est pas forcément compilée dans le même lot que celle qui s'y réfère. La désignation peut être soumise à une clause d'indexation par le "registre software" de numéro 0 à 7. En cours de descente le contenu du registre s'ajoute à la désignation explicite; la valeur du registre est prise soit lors du placement de la désignation dans la pile auxiliaire (option PILE), soit lors de l'extraction de la pile pour affichage (option AFFICHAGE). La clause MASQUE stipule que des sélections de la page doivent être masquées à l'affichage; l'énumération des choix masqués est soit explicite dans l'instruction successeur (suite de numéros logiques de sélections entre parenthèses), soit contenue dans un "registre software" examiné en cours de descente au moment de l'affichage; ces derniers masques ont les avantages et les inconvénients inhérents à leur nature dynamique. REPETITION produit la réinsertion automatique de la page successeur dans la pile (au-dessous des successeurs éventuels qu'elle-même désigne) sans que les sélections de cette page le demandent. La clause FANTOME provoque le traitement du choix numéro "position écran" sur la page successeur sans que cette page apparaisse au terminal pendant la descente. Cet artifice permet notamment de lever certaines restrictions, rarement

gênantes d'ailleurs (nombre de successeurs limité à sept , trois manipulations de registre possibles dans une sélection); il suffit de faire figurer ces opérations excédentaires dans une sélection de page fantôme, laquelle peut d'ailleurs fort bien appeler une page fantôme etc. Les autres clauses de l'instructions SUCCESSEUR servent en particulier à modifier partiellement l'affichage du successeur pour le rendre plus spécifique du contexte où il apparaît, à introduire dans le MBG une contribution préfixant celle qu'y apportera la sélection de la page appelée etc.

7. L'instruction

TASK-LIST <nom de chaîne>

demande le débranchement vers une chaîne de tâches d'application auxiliaire après que toutes les autres opérations requises par la sélection en cours ont été effectuées (mise à jour de MAC/MBG, manipulations de registre, chargement de la pile auxiliaire...). Lorsque la chaîne rend le contrôle à EXAC, la descente reprend avec l'envoi de la page désignée par le sommet courant de la pile auxiliaire.

La dernière des instructions signalées dans l'aperçu syntaxique 3.3.1,

DESTIN <nom de chaîne>

spécifie la chaîne de validation sémantique/exécution appelée en fin de descente; une telle instruction devrait figurer dans l'une au moins des sélections rencontrées au cours de toute descente; la dernière prescription de DESTIN traitée substitue son effet aux prescriptions antérieures éventuelles.

8. Pour terminer remarquons que la syntaxe et le compilateur DESPAC sont plus évolués que ne le laissent supposer les schémas syntaxiques que nous avons donnés au paragraphe 3.3.1 :

- après SELECTION, l'ordre des instructions dans une sélection donnée est indifférent;
- de même, les informations globales de page type-S peuvent apparaître à peu de chose près dans n'importe quel ordre;
- ceci est vrai aussi des options et clauses permises dans une instruction donnée, comme SUCCESSEUR...
- enfin, les instructions répétées dans plusieurs sélections peuvent être déclarées avant les prescriptions de sélections, précédées de l'énumération des numéros logiques de sélections auxquels tel lot d'instructions s'applique. Ces déclarations d'instructions dites "implicites" simplifient notablement l'écriture et l'entretien des pages.

3.4. Réalisation de DESPAC

Le langage utilisé dans le compilateur DESPAC est essentiellement FORTRAN; quelques routines sont écrites en assembleur CDC 6000/CYBER, COMPASS, pour accéder aux tables de la page en gestation et pour compacter ces tables (voir paragraphe 3.6).

Le développement de ce compilateur représente un travail d'environ 1 1/2 à 2 hommes-années.

Le produit est actuellement stable; la détection des erreurs de syntaxe est très rigoureuse. Un effort particulier a été fait pour éviter qu'une erreur de syntaxe fourvoie le compilateur au-delà d'une instruction, voire d'une clause, ou masque d'autres erreurs de syntaxe.

3.5. Entretien des librairies de pages

Les paragraphes 1.4.5 et 3.2 ont déjà signalé l'existence et le rôle de l'utilitaire CHAPAC. Donnons à son sujet encore quelques détails. Il permet :

1. de créer une nouvelle librairie sur un noyau de pages compilées initial, selon des directives spécifiant la taille (nombre de pages) de chacun des groupes de la librairie;
2. de mettre à jour une librairie existante par adjonction de nouvelles pages compilées et/ou remplacement de pages par des pages correspondantes recompilées;
3. de transformer une librairie existante en changeant la taille de ses groupes;
4. de fusionner en une, deux librairies existantes, l'une des librairies étant considérée comme primaire, l'autre comme secondaire, lorsque des pages de même identificateur se retrouvent dans les deux librairies source;
5. d'analyser une librairie, complète ou par groupes spécifiques, selon divers critères ou niveaux, d'ailleurs combinables dans une même analyse; ces niveaux sont définis comme suit :
 - références croisées des pages : étant donnée une page, l'ensemble des pages qui s'y réfèrent et selon quelles modalités (indexation, masque, etc.);
 - utilisation des registres software : étant donné un registre, l'ensemble des pages qui s'y réfèrent et selon quelles modalités (consultation, altération de contenu, opérande, etc);

- appel des chaînes d'application : étant donné le nom d'une chaîne, l'ensemble des pages qui l'appellent et selon quelle qualité;
- informations administratives et globales des pages, présentées selon deux tris : par noms source et par identificateurs compactés (paragraphe 3.2) des pages.

Seules les opérations 1 et 3 demandent des directives (tailles des groupes créés ou altérés); les autres se contentent de la carte de contrôle appelant CHAPAC et précisant les options voulues.

A propos de l'analyse de librairie de pages par CHAPAC, soulignons le fait qu'elle porte directement sur le fichier librairie des pages, tel qu'il est utilisé pour la descente dans l'arbre de capture; ses résultats sont ainsi garantis exempts de tout décalage entre ce qu'on analyse et le produit servant à l'exploitation, décalage que l'on pourrait toujours redouter si l'analyse portait sur des pages en format source.

3.6. Réalisation de CHAPAC

La réalisation de CHAPAC n'appelle que peu de commentaires :

1. Il est écrit moitié en PREFTN (précompilateur FORTRAN développé pour le projet Diogène [Mazzario 76]), moitié en assembleur COMPASS. L'usage fait de l'assembleur s'explique par le souci de centraliser la paramétrisation de CHAPAC, EXAC et, partiellement, DESPAC. La structure des pages compilées est ainsi décrite de manière unique pour les trois produits. Le choix fait de l'assembleur pour développer EXAC est justifié au chapitre suivant.
2. Du point de vue de sa structure en général, CHAPAC est formé d'un "overlay" de base et d'un ensemble d'overlays de premier niveau, dont l'un traite la carte de contrôle d'appel CHAPAC, et chacun des autres, l'une des options fondamentales : création, adjonction/remplacement, transformation, fusion, analyse de librairie. Cette dernière recourt à d'autres overlays, en particulier pour le tri du fichier de travail résultant de l'analyse, tri effectué par le module standard de tri/fusion fourni par le constructeur.
3. Le développement de CHAPAC représente un travail de 4 hommes-mois environ. Vu la variété restreinte des situations auxquelles il doit faire face cet utilitaire a présenté d'emblée une sécurité de fonctionnement satisfaisante.

4. Réalisation de la descente

La chaîne de tâches EXAC peut être envisagée de plusieurs points de vue :

- interactions avec l'opérateur au terminal,
- interactions avec les autres chaînes,
- mécanismes utilisés dans la chaîne,
- découpage en tâches et passage du contrôle de l'une à l'autre,
- codification des programmes de la chaîne.

Ce chapitre présente ces divers aspects.

4.1. Interactions avec l'opérateur

La démonstration du paragraphe 1.2 a permis d'illustrer la plupart de ces interactions.

4.1.1 Nature des messages physiques

Les messages physiques envoyés par EXAC au terminal sont uniquement des affichages renouvelant entièrement le contenu de l'écran; ils sont de deux types :

- affichages de pages type-S,
- affichages de pages type-C.

Il n'y a pas de différence fondamentale dans le mode d'envoi de ces affichages; dans les deux cas EXAC s'efforce de n'envoyer que les caractères significatifs, tirant au besoin parti de l'adressage du curseur du terminal pour éviter de transmettre des blancs entre les champs d'une même ligne, de manière à réduire le temps d'affichage. A 9600 bauds, taux de transmission disponible dans le projet, l'envoi d'un caractère prend 1 ms environ; tout gain dans ce domaine ne saurait être négligé. La différence principale, du point de vue des messages physiques, entre pages des types S et C, concerne le message entré au terminal par l'opérateur :

- "message de sélection" en présence d'un affichage type-S, contenant le caractère étiquetant le choix fait sur la page ou le symbole d'une touche fonction ("=" pour la rétrogradation, "*" pour le passage du mode normal en mode correction et vice versa);

- "Message Format Libre" ou MFL entré au clavier en présence d'un affichage type-C; le contenu exact dépend en particulier de certaines caractéristiques du terminal utilisé; en règle générale le MFL contient, au départ du terminal, la suite des champs non-protégés transmissibles définis sur l'affichage, séparés les uns des autres par des caractères de contrôle.

En délimitant de façon rigoureuse les interfaces voulues, l'essentiel d'EXAC, et a fortiori les applications, sont rendus indépendants du choix final d'un terminal interactif pour le projet Diogène, choix retardé le plus possible avant l'entrée en production pour des raisons technico-économiques.

4.1.2 Origine des affichages

Chacun des affichages envoyés par EXAC est tiré d'une page de la librairie. Avant envoi au terminal il est soumis à certains traitements (abstraction faite des ultimes conversions de code et élimination des blancs s'il y a lieu) :

- remplissage de sélections ou de champs par des informations reçues d'une chaîne de remplissage s'il s'agit d'une page squelette (paragraphe 1.4.4);
- altération d'une ligne de commentaire par le contenu d'un format spécifié éventuellement dans l'instruction successeur appelant cette page;
- masque appliqué à certains choix;
- inclusion éventuelle du MAC;
- inclusion des codes de contrôle placés en haut à gauche de l'affichage pour indiquer l'état de la descente en cours, signaler une page clef ou squelette, etc.

Finalement, malgré le caractère relativement statique des pages de l'arbre, il est rare qu'une même page donne lieu à deux affichages rigoureusement identiques.

4.1.3 Affichages spéciaux

Dans leur majorité, les affichages envoyés sont ceux de pages désignées comme successeurs dans les sélections faites antérieurement dans la descente. Toutefois, certaines pages sont appelées par EXAC lui-même sans être jamais désignées explicitement comme successeurs par les pages que fournissent les rédacteurs de pages. Ces affichages spéciaux, désignés par EXAC1, EXAC2, ..., EXAC8, sont apparus, du moins les plus significatifs d'entre eux, EXAC1, EXAC2 et EXAC4, dans la démonstration du paragraphe 1.2.

Les pages correspondant à ces affichages sont placées automatiquement par CHAPAC dans toute librairie de page lors de sa création dans le groupe 0 réservé à cet usage. Bien que ces affichages aient les apparences de pages ordinaires, le traitement des sélections qu'ils présentent est tout à fait spécifique, autrement dit ne relève pas du schéma général de l'interprétation d'une sélection donnée au paragraphe 1.4.6. Il en sera question lors de l'examen des mécanismes utilisés dans la descente.

4.1.4 Envoi des affichages EXACi

Voici résumées les circonstances où sont envoyés les affichages spéciaux EXAC1, ..., EXAC8.

EXAC1 : conditions anormales détectées dans une chaîne d'application (auxiliaire, remplissage, validation MFL); une telle chaîne peut demander à EXAC l'envoi d'EXAC1 avec un diagnostic.

EXAC2 : demande de visa après validation sémantique/exécution; cet affichage contient le message, un diagnostic éventuel et des choix que la chaîne de validation sémantique peut masquer à volonté; par exemple le choix "OK" sera masqué si l'application entend empêcher la mise à jour de la banque de données.

EXAC3 : descente terminée sans production d'un nouveau message de transaction; intervient en général à l'enchaînement de transactions si l'opérateur choisit de sortir sur toutes les pages clefs qui lui sont présentées.

EXAC4/5 : présentation éclatée du message permettant à l'opérateur d'indiquer une portion de message à corriger; EXAC4 apparaît sur activation de la touche *. EXAC5 présente la suite du message si EXAC4 ne peut le montrer en entier.

EXAC6/7 : interruption de descente automatique ou fictive provoquée par l'une ou l'autre des circonstances précisées aux paragraphes 4.3.8 et 4.3.5.

EXAC8 : erreur fatale intervenant dans EXAC lui-même ou dans une chaîne d'application. Dans ce dernier cas, EXAC est rappelé comme liquidateur standard d'erreur.

4.2. Interactions avec d'autres chaînes

Les informations constituant l'interface entre EXAC et les applications sont passées par le bloc de communication. La nature et la structure de ces informations dépendent du sens de transfert de contrôle considéré.

4.2.1 Une chaîne d'application appelle EXAC

EXAC reçoit le contrôle dans une tâche d'interface unique, opérant des débranchements dépendant essentiellement d'un code transmis dans le bloc de communication, caractéristique du genre d'action que l'application attend d'EXAC. Les valeurs de ce code, symbolisées par COD11, COD12, ..., COD20, vont désigner les classes d'appel d'EXAC décrites ici brièvement :

COD11/COD12 : mise en route de la descente sur une page identifiée soit explicitement dans l'interface, soit par une convention faisant appel par exemple au nom du terminal connecté ou au contenu du premier message reçu.

COD13 : reprise de la descente au retour d'une chaîne auxiliaire ou de validation MFL.

COD14 : renvoi d'une page type-C pour correction des informations introduites par l'opérateur et validées par une chaîne de validation MFL. L'interface prévoit le passage d'un diagnostic à exhiber sur la page et de codes modifiant le mode d'affichage de certains champs pour mettre en évidence (clignotement par exemple) une zone suspecte.

COD15 : reprise de la descente au retour d'une chaîne auxiliaire ou de validation MFL, avec demande d'adjonction d'un complément aux messages MAC/MBG.

COD16 : enchaînement de transactions après une chaîne de validation sémantique/exécution n'impliquant pas l'envoi d'EXAC2 pour demande de visa.

COD17 : envoi d'EXAC1 au retour d'une chaîne auxiliaire, validation MFL ou remplissage, pour signaler une condition anormale, en général non fatale.

COD18 : enchaînement de transactions après une chaîne de validation sémantique/exécution précédée de l'envoi d'EXAC2.

COD19 : retour d'une chaîne de remplissage de page squelette type-S.

COD20 : retour d'une chaîne de remplissage de page squelette type-C.

4.2.2 EXAC appelle une chaîne d'application

Quel que soit le type de chaîne appelée, EXAC donne dans des adresses fixes du bloc de communication :

- un code spécifiant le type de chaîne : remplissage-S ou C, auxiliaire, validation MFL, etc;
- un pointeur au MBG et sa taille courante;
- d'autres informations, comme le contenu des registres software.

L'information principale est en général le message MBG, syntaxé ACAP (paragraphe 1.4.7). Dans le cas des chaînes de validation MFL, une autre information essentielle est naturellement le MFL, résultant de la concaténation des informations entrées par l'opérateur dans les zones réceptrices de la page type-C appelant cette chaîne.

4.3. Mécanismes intervenant pendant la descente dans l'arbre

Ce paragraphe décrit les principaux mécanismes utilisés par EXAC pendant la descente dans l'arbre de capture. Au préalable il donne quelques indications sur l'utilisation du CB par EXAC.

4.3.1 Allure du CB sous EXAC, structure générale des messages

4.3.1.1 Schéma du CB (bloc de communication)

Le CB contient les informations permettant le découpage d'EXAC en tâches, et le passage de contrôle de la chaîne EXAC aux chaînes d'application et vice et versa. Cette portion de mémoire commune, dont la taille peut varier en principe d'une chaîne à l'autre, est, pour fixer les idées, de 512 mots dans les chaînes existantes (mots de 60 bits). Vue d'EXAC son allure est celle de la figure 4-1.

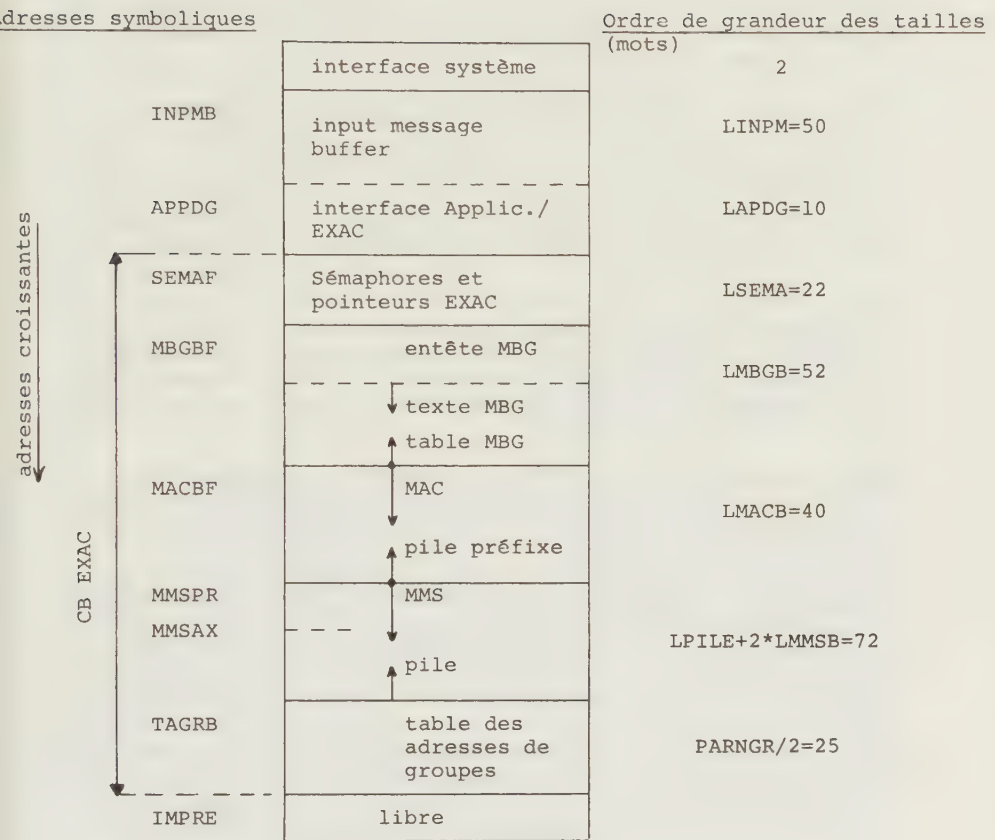


Fig. 4-1 Allure du CB vu d'EXAC

Commentaires sur la figure 4-1 :

INPMB (input message buffer, zone ainsi nommée pour des raisons historiques) : reçoit le code étiquetant le choix fait sur une page type-S ou le MFL (après sa conversion à partir du message physique reçu du terminal) entré sur une page type-C.

APPDG (interface application/EXAC) : ces quelques mots d'adresse fixe servent de base à l'interface EXAC/applications (et inversement); ils contiennent surtout des pointeurs aux informations, variables de cas en cas, passées par les applications à EXAC et réciproquement. Ces informations doivent être placées dans le CB en dehors de la zone touchée par EXAC ("CB EXAC" sur la figure).

SEMAF (sémaphores, pointeurs, etc) : variables, pointeurs et autres drapeaux contrôlant le déroulement d'EXAC; cette zone contient aussi les valeurs initiales et courantes de variables d'état comme les registres software, la prescription du nom de la chaîne de validation sémantique appelée en fin de descente, etc.

MBGBF (entête MBG, texte MBG, table MBG) : l'entête MBG prolonge en fait la zone précédente; toutefois la définition en est plus rigide dans la mesure où les applications y accèdent le cas échéant pour décoder le texte MBG (avoir son adresse de départ et sa taille courante); le texte MBG est le MBG proprement dit, syntaxé ACAP. La table MBG (de 1 mot par entrée) sert au découpage du MAC et du MBG en vue de correction par envoi d'EXAC4, de présentation partielle du MAC en rétrogradation, etc. Chaque contribution au MAC et/ou au MBG donne lieu à une entrée dans la table.

MACBF (texte du MAC, pile préfixe) : le texte du MAC ne demande pas de commentaire si ce n'est pour redire que son découpage éventuel est possible grâce à la table MBG; la "pile préfixe" (dont le rapport avec le MAC est de pur voisinage) est faite d'entrées de taille variable contenant :

- une contribution (préfixe) future au MBG définie dans une instruction successeur appelant une page;

et/ou

- un texte variable substitué à une ligne de commentaire d'une page appelée, texte spécifié par l'instruction successeur visant cette page.

L'accès à la pile préfixe est gouverné par les entrées de taille fixe de la pile des désignations de page (voir ci-dessous). Toute désignation de successeur implique une entrée dans la pile proprement dite, alors que l'entrée dans la pile préfixe n'intervient que selon les nécessités.

MMS : message gardant la trace de la descente à raison d'un mot par entrée, chaque entrée correspondant au passage par une page; s'il s'agit d'une page type-S on garde en particulier le numéro séquentiel de la page en librairie, la position sélectionnée sur l'écran, le "numéro de modification" de la page permettant d'assurer l'identité des pages squelettes et dynamiques utilisées en descentes réelles et fictives ou la détection de toute divergence.

pile : pile auxiliaire de désignation des pages candidates à l'affichage; chaque entrée a deux mots contenant le numéro séquentiel de la page en librairie (sous réserve de modification par une indexation éventuelle), le masque applicable (explicite ou référence à un registre), les relations voulues avec le MMS, la pile préfixe etc.

TAGRB (table des adresses de groupes) : table permettant de passer de l'identificateur compacté d'une page à son numéro séquentiel en librairie (paragraphe 3.2); la table est lue de la librairie (page spéciale de numéro zéro) pour la mise en route de la descente.

zone CB EXAC : dans son ensemble c'est la zone de travail d'EXAC; cette portion du CB est sauvée et récupérée par EXAC selon modalités décrites au paragraphe 4.3.4.

IMPRE : adresse de départ d'une zone libre du CB, dans laquelle EXAC n'écrit pas et ne va chercher des informations que si l'interface applications/EXAC l'y invite (adresses spécifiées par l'application dans les mots de base de l'interface, APPDG). Les adresses de base des diverses zones (APPDG, SEMAF, MBGBF...) sont des paramètres d'assemblage d'EXAC. Six entités se partagent trois zones (texte MBG/table MBG, MAC/pile préfixe, MMS/pile) en croissant par paires l'une à l'encontre de l'autre. On économise ainsi de la mémoire. On en gagnerait encore en rendant dynamiques les adresses de base de ces trois zones et en relogeant, s'il y a lieu (risque de collision de deux entités d'une paire donnée) une paire ou l'autre, en cours de descente. Jusqu'ici les problèmes de place mémoire n'ont pas été assez aigus, ni les gains réalisables, assez substantiels, pour justifier ce développement supplémentaire.

4.3.1.2 Relations entre messages

Les relations établies entre entrées correspondantes des messages et piles (sous forme de pointeurs relatifs) jouent un rôle essentiel dans les mécanismes décrits aux paragraphes suivants de ce chapitre.

Il y a un certain arbitraire dans la division des messages MMS, MAC, MBG. En principe on aurait pu produire un seul message physique dont chaque entrée aurait comporté un aspect MMS, MAC, MBG. Toutefois, l'exploitation du MBG, compte tenu de la syntaxe ACAP, se fait mieux avec un texte continu; le même argument s'applique évidemment au MAC; le MMS s'exploite plus commodément en cours de recul ou de descente fictive s'il a des entrées de taille fixe; enfin la table MBG est séparée du texte MBG pour que le texte MBG soit continu, mais n'est pas incorporée au MMS parce que l'existence d'une entrée MMS n'implique pas celle d'une entrée MBG et parce que, dans certains cas, le fait d'avoir un MMS aussi réduit que possible, est un avantage décisif (recherche de convergence des chemins après correction, en vue d'une descente automatique, paragraphe 4.3.8.2).

La figure 4-2 schématise les relations existant entre les messages et piles.

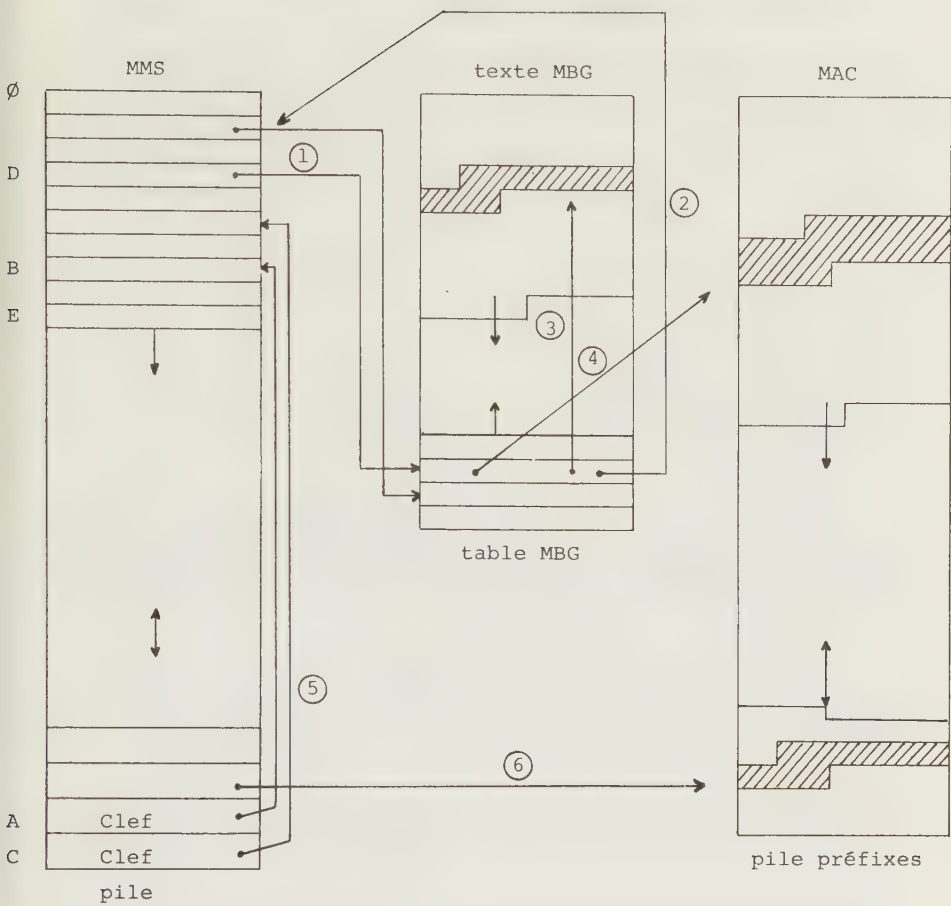


Fig. 4-2 Relations entre messages

4.3.2. Descente normale

4.3.2.1. En présence d'une page type-S

On suppose que l'opérateur choisit une position effectivement sensible; le traitement de cette sélection consiste pour EXAC à examiner le contenu de la table des informations de contrôle de sélection et, en fonction de ce contenu, à effectuer les opérations suivantes :

- a) traitement d'une option SORTIE éventuelle;
- b) débranchement éventuel sur une TL* abandon (si TERMINAISON ABANDON);
- c) manipulations de REGISTRES;
- d) prise en compte d'une spécification de TL de validation sémantique (instruction DESTIN);
- e) adjonction éventuelle au MBG (instr. INFO-BG) en tenant compte de clauses REPETITION/PREFIXE agissant peut-être sur la présente page du fait de l'instruction SUCCESSEUR qui l'a appelée; mise à jour du compteur de parenthèses ouvertes à ce niveau de pile;
- f) adjonction éventuelle au MAC (instructions INCLUSION MAC et, éventuellement, REMPLACEMENT);
- g) prise en compte d'une spécification de débranchement facultatif sur une page type-C (instr. CLICHE-MFL);
- h) mise à jour du MMS (pour l'essentiel une entrée MMS contient le numéro de la page où est faite la sélection et le numéro de la position sensible sélectionnée);
- i) chargement de la pile et, s'il y a lieu, de la pile auxiliaire des préfixes (instructions SUCCESSEUR); en l'absence de successeurs désignés par cette sélection, fermeture dans le MBG de délimiteurs ouverts à ce niveau dans la pile et annulation du compteur;
- j) débranchement sur une TL auxiliaire éventuelle (instruction TASK-LIST);
- k) débranchement sur la TL de validation sémantique s'il y a "TERMINAISON" imposée par cette sélection;
- l) sondage de la pile (débranchement sur la TL de validation sémantique prescrite antérieurement dans la descente, si la pile est vide ou si la prochaine entrée pile est assortie du sémaphore clef; .

* Dans la suite du paragraphe 4.3 l'abréviation TL (Task-List) désigne une chaîne de tâches d'application.

- m) lecture de la page du sommet de la pile;
- n) débranchement vers une TL de remplissage s'il s'agit d'une page squelette (instruction REMPLISSAGE), ou vers le traitement de page type-C s'il y a lieu;
- o) préparation de l'envoi de l'affichage (masque éventuel, etc.);
- p) requête SENDM pour l'affichage;
- q) requête RECEIVE pour attendre la réponse de l'opérateur;
- r) analyse de sa réponse et s'il s'agit d'une vraie position sensible aller en a). Autrement, traitement de fonctions particulières : recul, correction ponctuelle de message, débranchement sur un cliché MFL facultatif (possible si, au traitement de la sélection antérieure, le point "g" s'appliquait).

4.3.2.2. En présence d'une page type-C

- a) examen du dernier caractère du MFL entré par l'opérateur; s'il s'agit d'une touche fonction, elle est traitée comme telle;
- b) s'il s'agit effectivement d'un MFL, le MAC et le MBG sont complétés (instruction INCLUSION-MFL);
- c) le MMS est complété (entrée contenant le numéro de page en librairie et un sémaphore signalant que c'est une page type-C);
- d) s'il y a lieu, débranchement sur une TL de validation MFL (instruction VALIDATION);
- e) si la page type-C sert de court-circuit à une descente par sélections (instruction ECREMAGE), les entrées supérieures de la pile sont éliminées, jusqu'à celle de la page visée par l'instruction ECREMAGE. Les parenthèses ACAP ouvertes au niveau des entrées éliminées sont fermées dans le MBG;
- f) si la page termine la descente (instruction TERMINAISON), l'effet est le même que précédemment, mais l'écrémage de la pile se poursuit jusqu'à ce que la pile soit vide ou qu'apparaisse une entrée de page clef.

4.3.3. Mise en route de la descente

La mise en route de la descente dans l'arbre peut être demandée de deux manières (voir paragraphe 4.2.1 COD11/COD12) :

- a) page initiale prescrite par l'application appelante
- b) page initiale déduite du contenu du message de communication trouvé dans "l'input message buffer"; ce cas se subdivise en deux :
 - b1. message initial envoyé par la centrale de commutation reliant téléphoniquement le pool des opératrices Diogène à l'ensemble des unités de soins (téléstations) de l'Hôpital;
 - b2. autre message initial.

Dans tous les cas, avant la mise en route de la descente, la zone des pointeurs et sémaphores ainsi que les messages et piles sont initialisés. EXAC conserve l'état initial des registres software et des spécifications de niveau, interdictions, TL de validation sémantique, en vue de réinitialisations ultérieures de la descente (pour descentes fictives ou choix RECOMMENCER sur affichages EXACi).

4.3.4. Débranchement vers des TL extérieures à EXAC

EXAC se débranche vers une TL qui lui est extérieure (ou TL d'application) soit sur demande explicite, soit en raison de l'état de la pile, soit par un choix sur EXACi.

Les divers cas de débranchement ont été donnés au paragraphe 4.3.2 de sorte qu'il ne reste qu'à préciser quelques traits communs à tous ces débranchements :

- a) Les TL d'applications se situent au même niveau qu'EXAC; celui-ci ne peut forcer une TL à lui rendre le contrôle; tout au plus peut-il espérer qu'elle le lui rende dans de bonnes conditions (de CB) si la descente doit se poursuivre.
- b) Sur demande, au niveau source DESPAC par l'option SAUVE-CB, dans l'instruction TASK-LIST par exemple, EXAC sauve sur disque la portion de CB (zone CB EXAC dans la figure 4-1) qui lui permettra de reprendre correctement la descente en récupérant cette portion de CB au retour de la TL appelée; ceci vaut pour les TL auxiliaires et TL de validation MFL. Aucun sauvetage n'est prévu pour les TL de REMPLISSAGE. Le sauvetage est automatique avant l'appel de la TL de validation sémantique/exécution; la récupération a lieu, en général, au traitement du choix fait sur EXAC2, pour permettre l'enchaînement de transaction ou la correction de message.

Le sauvetage de CB EXAC, fait sur un fichier local, sorte d'extension de la mémoire en cette occurrence, n'est pas une mise à jour de la banque de données du point de vue des mécanismes de sauvegarde et reprise.

4.3.5. Descente fictive

4.3.5.1 Définition et but

Reprenons les définitions données au début du paragraphe 1.4.8 :

La descente fictive, ainsi nommée parce qu'elle se fait sans intervention de l'opérateur, est le mécanisme de base utilisé dans la correction de message (soit par recul soit ponctuelle) et l'enchaînement de transactions par pages "clef" ou "monitoring". De façon générale, le but recherché en effectuant une telle descente fictive est un repositionnement dans l'AC en un point A où l'on est passé antérieurement dans une descente réelle, avec rétablissement exact des conditions dans lesquelles on se trouvait en ce point A lors de la descente réelle. Les conditions à reconstituer sont :

- la pile (et la pile préfixe auxiliaire) y compris les compteurs de parenthèses MBG ouvertes aux divers niveaux;
- l'état des registres software;
- l'état des interdictions, des prescriptions de TL abandon, TL validation sémantique et d'affichage de repos.

La descente fictive permet d'atteindre ce but en refaisant la descente, de l'origine de l'arbre au point A, sans intervention de l'opérateur grâce à la trace gardée dans le MMS. Naturellement on peut imaginer d'autres méthodes que la descente fictive pour obtenir le même effet et qui éviteraient, peut-être, la relecture des pages que nécessite la descente fictive. Toutefois, la descente fictive offre deux avantages décisifs :

- elle permet de reconstituer les conditions en A en ne gardant dans le CB qu'une trace minimum (1 entrée MMS de 1 mot par pas dans la descente réelle);
- les routines de traitement utilisées en descente fictive sont pour l'essentiel celles qui servent dans une descente réelle, ce qui simplifie la conception et l'entretien des programmes.

4.3.5.2 Obstacle en cours de descente fictive; réexécution des TL auxiliaires

La descente fictive implique toutefois certaines limitations dans les possibilités de repositionnement sans intervention de l'opérateur, limitations discutées ici sur la base de deux définitions :

Obstacle :

Interaction avec l'opérateur du terminal, autre qu'une sélection sur une page type-S ordinaire ("ordinaire" signifiant : page qui n'est pas un affichage spécial EXACi).

Exemples d'obstacles :

- a) TL auxiliaire interagissant directement avec l'opérateur,
- b) TL auxiliaire rendant le contrôle à EXAC avec demande d'envoi de l'affichage EXAC1,
- c) page type-C appelant ou non une TL de validation.

Une TL auxiliaire de type a) doit se signaler comme telle à EXAC par l'interface EXAC/applications. Les autres obstacles sont détectables par EXAC.

Changement d'état :

Altération du contenu des registres software, des interdictions et autres conditions d'état énumérées au paragraphe 4.3.5.1., provoquée par une TL auxiliaire ou TL de validation MFL, altération demandée par cette TL lorsqu'elle redonne le contrôle à EXAC. Ces changements d'état sont en principe détectables par EXAC si l'interface Applications/EXAC est utilisé correctement mais une TL peut aussi signaler explicitement qu'elle produit un changement d'état.

Le fait pour une TL auxiliaire ou de validation d'apporter un complément de message MAC ou MBG n'est pas un "changement d'état".

Le MMS garde la trace (sous forme de deux sémaphores) de la présence d'obstacles et changements d'états. En cours de descente fictive on tourne autour des obstacles qui ne sont pas combinés à un changement d'état.

Conséquences :

- 1) Une TL auxiliaire interactive n'est pas réexécutée en descente fictive.
- 2) Une TL de validation MFL n'est pas réexécutée en descente fictive (sa réexécution exigerait qu'on dispose à nouveau du MFL).

- 3) Une TL auxiliaire non interactive est réexécutée en descente fictive si elle ne demande pas l'envoi d'EXAC1.

La descente fictive ne peut se poursuivre au-delà d'un obstacle combiné à un changement d'état; en effet, si l'obstacle était tourné les conditions énumérées au paragraphe 4.3.5.1. ne seraient pas intégralement reconstruites lorsqu'on atteindrait le point A visé par la descente fictive; or l'obstacle, s'il n'est pas tourné, implique l'intervention de l'opérateur, intervention qui, en général, n'a pour l'opérateur, aucun rapport apparent avec la manipulation qu'il veut effectuer, par exemple une correction de message. C'est la raison pour laquelle, lorsqu'il trouve pendant la descente fictive une entrée MMS signalant un obstacle non contournable EXAC envoie à l'opérateur l'affichage spécial EXAC7 signalant l'interruption de descente fictive (voir paragraphe 4.1.3). De ce point, la descente doit être poursuivie "à la main" par l'opérateur.

4.3.5.3. Autres causes d'interruption de descente fictive

La descente fictive peut devoir être interrompue pour d'autres raisons :

- a) Les numéros de modification de pages dynamiques et ceux des informations de remplissage des pages squelettes sont gardés dans les entrées MMS correspondant au passage par ces pages. Pendant la descente fictive EXAC s'assure que le numéro de modification dans l'entrée MMS concorde avec celui de la page lue ou remplie en descente fictive; en cas de divergence la descente fictive s'interrompt avec envoi d'EXAC7.
- b) Des circonstances (très improbables) peuvent nécessiter une interruption avec envoi d'EXAC7 :
 - b1. collision entre la portion du MMS atteinte en descente fictive et une hauteur atteinte par la pile antérieurement dans cette descente fictive (ce qui ferait que le MMS, ayant été écrasé, il ne permet plus de guider la descente fictive); se référer au schéma du paragraphe 4.3.1.1. qui montre que le MMS et la pile occupent une zone mémoire commune. EXAC mémorise la hauteur maximum de la pile pour s'assurer de l'intégrité de la portion de MMS utilisée;
 - b2. collision entre une portion non régénérable du MAC (non régénérée à cause d'un obstacle à tourner) et une hauteur atteinte antérieurement dans la descente fictive par la pile auxiliaire des préfixes; le schéma de la figure 4-1 montre que cette pile préfixe et le MAC occupent une zone mémoire commune. EXAC mémorise la hauteur maximum de la pile préfixe pour s'assurer de l'intégrité des portions du MAC non régénérées.

4.3.6. Enchaînement de transactions

4.3.6.1 Causes de l'enchaînement et méthode utilisée

L'enchaînement de transactions peut être demandé par une TL ayant exécuté une transaction et rendant le contrôle avec COD16 ou, et c'est le cas normal, par un choix fait sur EXAC2.

EXAC procède alors comme suit (figure 4-2) :

- a) examen de la pile résiduelle; si elle est vide, envoi d'EXAC3; sinon, supposons que le sommet soit l'entrée pile A où le sémaphore "clef" est posé;
- b) la page ainsi désignée est envoyée au terminal;
- c) deux cas peuvent se présenter :
 - c1. L'opérateur choisit une sélection où ne figure pas l'option SORTIE, EXAC fait une descente fictive en suivant le MMS, de l'origine $\emptyset$ à l'entrée précédant B puis interprète la sélection faite par l'opérateur sur la page clef; ainsi le processus normal de descente réelle est réamorcé.
 - c2. L'opérateur choisit une sélection où figure l'option SORTIE; EXAC examine la prochaine entrée dans la pile; supposons que ce soit C où le sémaphore "clef" est posé ceci nous ramène en a), le pointeur 5 désignant la nouvelle limite de la descente fictive éventuelle.

4.3.6.2 Limitation

Lors de l'enchaînement de transactions, la pile résiduelle ne doit contenir que des entrées assorties du sémaphore "clef". EXAC ressent tout autre situation comme une erreur car elle reviendrait à permettre qu'une descente puisse agir sur une autre alors que l'enchaînement de transactions par page clef est conçu uniquement comme une facilité offerte à l'opérateur pour générer plus rapidement des messages de transactions et non comme un moyen d'imposer à l'opérateur une séquence donnée de transactions.

4.3.7. Rétrogradation

La touche fonction "=" permet, en général, de rétrograder. Supposons que l'opérateur décide de rétrograder alors que la dernière entrée MMS créée est l'entrée D (figure 4-2); EXAC examine l'entrée D, lit la page correspondante et l'affiche en tenant compte de la taille du MAC qui avait été produit à ce niveau-là (information accessible via

les pointeurs 1 et 4). Sur la page ainsi présentée, l'opérateur peut :

- a) activer à nouveau "="; EXAC recule d'une entrée dans MMS, affiche la page correspondante, etc.;
- b) effectuer une sélection (s'il s'agit d'une page type-S) ou entrer un message au clavier (page type-C). Dans ce cas EXAC fait une descente fictive jusqu'au point atteint par rétrogradation, interprète la sélection ou message déjà reçu sur la dernière page présentée en rétrogradation; ainsi le processus normal de descente réelle est réamorcé;
- c) en général (s'il n'est pas parti en rétrogradation sur un affichage EXACi), l'opérateur peut aussi activer la touche "*" ce qui lui permet de reprendre la descente au point d'où il a rétrogradé, comme s'il n'avait pas reculé. EXAC fait en sorte que la rétrogradation perturbe assez peu le CB pour que la reprise de la descente puisse se faire sans problème.

Diverses considérations conduisent à certaines restrictions en matière de rétrogradation :

- en cours de rétrogradation les masques agissant sur les pages en descente réelle ne sont pas appliqués (puisque le MMS n'en garde pas trace) ce qui ne signifie pas qu'il soit possible de repartir en marche avant sur une position qui devrait être masquée; en effet, la descente fictive reconstruit les masques et si la position choisie est en fait masquée on se trouve dans la situation où l'on choisirait, en descente réelle, une position sensible masquée.
- en cours de rétrogradation les pages squelettes ne sont pas présentées (faute, en particulier, de pouvoir appeler valablement une TL de remplissage alors que l'état du CB n'est pas celui qui prévaut en descente réelle).
- comme indiqué à l'alinéa "c", si l'on passe en rétrogradation en présence d'un affichage EXACi, la touche fonction "*" est inhibée car l'affichage EXACi est présenté, en descente réelle, dans des conditions qui ne sont pas exactement celles d'une page normale.
- en présence d'une page clef présentée pour l'enchaînement de transactions la touche fonction "=" est inhibée.

Dans l'ensemble ces restrictions ne limitent que fort peu les possibilités de rétrogradation.

4.3.8. Correction ponctuelle; descente automatique

La touche fonction "*" permet en général de passer d'une descente en mode normal au mode correction et inversement. Soit E (figure 4-2) la dernière entrée MMS créée en descente normale réelle. L'opérateur du terminal active la touche "*". EXAC prépare l'affichage EXAC4 dont les choix contiennent les contributions successives au MAC apportées en cours de descente; ceci est possible grâce aux entrées de la table MBG via le pointeur 4. Lorsque l'opérateur a répondu, la sélection faite sur EXAC4 peut donc être associée à une entrée de la table MBG qui, à son tour, grâce au pointeur 2, définit une entrée MMS, D par exemple. Une descente fictive Ø-D permet de se positionner sur l'affichage où a été faite la sélection générant la portion de message à corriger. L'opérateur fait un nouveau choix sur la page présentée ce qui amorce la descente réelle.

Cette descente réelle se fait en mode correction, distinct du mode normal sur les points suivants :

- avant de passer en descente réelle, si l'espace entre MMS et pile le permet, EXAC crée une copie témoin du MMS;
- pendant la descente réelle, EXAC examine si la descente réelle rejoint la descente antérieure, par référence au MMS témoin. S'il y a raccordement, EXAC propose à l'opérateur, par le caractère "*" placé en haut à gauche de l'écran, de poursuivre lui-même la descente jusqu'à l'entrée E du MMS. L'opérateur l'accepte par la touche "*"; la descente faite alors par EXAC est dite "automatique"; elle ressemble beaucoup à la descente fictive du paragraphe 4.3.5, mais s'en distingue toutefois :
 - a) par la régénération totale de MAC et MBG qui exclut de contourner un "obstacle" (paragraphe 4.3.5.2) lequel provoque l'interruption de la descente automatique, avec envoi d'EXAC6;
 - b) par un test de "divergence" par rapport au chemin antérieur; en effet, lorsqu'EXAC détecte la convergence des chemins antérieurs et corrigés ceci n'empêche pas que ces chemins puissent diverger ultérieurement vu que les conditions lors de la convergence ne sont pas garanties identiques à ce qu'elles étaient antérieurement; en particulier des registres peuvent avoir des valeurs différentes du fait de la correction; EXAC teste donc pendant la descente automatique qu'une

sélection donnée sur une page donnée produit bien l'apparition de la même page que celle du chemin antérieur. Si ce n'est pas le cas, l'affichage EXAC6 est envoyé à l'opérateur. Ce même affichage est envoyé si une sélection faite pendant la descente réelle est devenue masquée en descente automatique de par la correction (masque par registre contenant une valeur différente de celle qu'il avait en descente réelle);

- c) par le fait que toute page squelette est considérée comme un obstacle : elle peut être remplie de manière différente pendant la descente automatique, de sorte que la position sélectionnée lors de la descente réelle ne peut servir sans risque.

4.4. Tâches constitutives d'EXAC

4.4.1 Contraintes agissant sur le découpage en tâches

Des mécanismes décrits au paragraphe précédent on peut dégager certains thèmes fonctionnels donnant quelques indications sur un découpage possible d'EXAC en modules. Que les modules définis par une telle analyse doivent correspondre à des tâches physiques distinctes n'est pas évident; il faut en tout cas tenir compte des contraintes qu'imposent la recherche d'une efficacité (temps de réponse) satisfaisante en cours de descente. Nous allons énumérer ces contraintes et en préciser l'impact.

Dans l'état actuel du projet Diogène, chaque terminal connecté, lorsqu'il est réellement actif, est affecté à un "point de contrôle" distinct au sens du système d'exploitation NOS [CDC 76]. Ceci signifie, en particulier, que si N terminaux recourent simultanément à la même tâche A, le code de la tâche A se retrouve à N exemplaires, en mémoire centrale, ou en mémoire de masse, au hasard de la charge du système, des interactions avec les terminaux etc. Une contrainte de base pour EXAC est de faire que la taille de ses tâches, du moins de celles qui sont les plus demandées, soit aussi réduite que possible pour faciliter la cohabitation de plusieurs points de contrôle EXAC, en mémoire centrale, et pour diminuer les temps de transfert de la mémoire centrale vers la mémoire de masse et réciproquement (swap -out et -in). Dans le même temps, mais à l'encontre de la tendance précédente, il faut qu'une tâche en fasse assez pour n'avoir pas un passage de contrôle incessant de l'une à l'autre dans la chaîne.

A cet égard l'attention s'est portée principalement sur le code requis par la descente normale sans accident dans des pages de type-S, autrement dit, une suite de sélections faites sans que soient appelées des chaînes d'application (de remplissage, de validation MFL ou auxiliaires), et en l'absence de toute correction du message induite par l'opérateur. Dans une telle descente, le compromis adopté se caractérise par l'usage d'une seule tâche exécutant la boucle normale décrite au paragraphe 4.3.2.1, dont le code occupe moins de 4k (octal) mots de 60 bits.

Une autre contrainte imposée à la descente normale est de n'impliquer qu'un seul accès physique à la banque de données par pas dans la descente; c'est bien le cas dans la solution utilisée : il s'agit de lire la page désignée par le sommet de la pile auxiliaire; or la structure de la librairie des pages (paragraphe 3.2) garantit que toute page repérée par son identificateur compacté passé au travers de la table des adresses de groupes est amenée en mémoire centrale par un accès physique à la mémoire de masse.

Un dernier point auquel il importe de prendre garde est la taille de la portion de CB dévouée à EXAC, taille qui doit être raisonnable; selon la figure 4-1, cette portion, "CB EXAC", est de l'ordre de 200 mots (sur 512) dont moins de 150 peuvent être considérés comme spécifiques de la méthode de l'arbre de capture (puisque 50-100 mots sont pris par le message de transaction sous ses formes MAC/MBG, qui devrait bien exister d'une manière ou d'une autre, quel que soit le mode de saisie des messages).

4.4.2 Rapport avec l'économie de la prise de messages

L'examen de la dernière des contraintes mentionnées rejoint les préoccupations d'économie du paragraphe 2.1, et, de façon plus précise, le deuxième des aspects qui s'y trouvent évoqués, celui du matériel et du temps machine dévolus à la constitution des messages. Sans quitter le domaine des évaluations très qualitatives, on peut dire que le prix payé pour la prise de messages par arbre de capture, par rapport à une saisie de messages plus libre, mais avec les inconvénients indiqués au paragraphe 2.1, se décompose de la manière suivante :

1. intensification du trafic des messages physiques, du système central vers les terminaux : en général, il y a renouvellement du contenu de l'écran à chaque pas de la descente dans l'arbre, alors que dans un système moins contrôlé, on enverrait un petit message de temps en temps;

2. sollicitation très fréquente du module de prise des messages EXAC : à chaque pas dans la descente, EXAC intervient pour décider de la prochaine page à envoyer, alors que dans un système plus conventionnel, l'interface pourrait attendre d'avoir un message de transaction plus ou moins complet avant toute intervention significative; de cette sollicitation particulièrement fréquente découlent les contraintes imposées à EXAC pour permettre une descente normale tout de même efficace. Il est d'ailleurs clair que cette fréquence de sollicitation est contraignante aussi, si ce n'est davantage, pour le système d'exploitation dans son ensemble, puisqu'il est mis à contribution à chaque sélection pour relancer la partition attachée au terminal;
3. lourdeur relative de la mise en oeuvre d'un programme d'application spécifique d'une situation donnée en cours de prise de message; dans une prise de message où interface avec l'extérieur et application sont plus ou moins confondues, la validation partielle du message en cours de formation ne donne lieu à aucune surcharge liée au simple passage de contrôle d'un module à un autre, alors que dans l'arbre de capture, toute intervention des applications en cours même de descente consiste en l'appel d'une chaîne de tâches extérieure à EXAC (chaîne de remplissage, de validation MFL, etc.), suivi du retour dans EXAC, d'abord par la tâche commune d'aiguillage (voir ci-dessous), puis une tâche ou l'autre d'EXAC selon la situation rencontrée.

Si l'on examine de près cette "facture", on se rend compte que son troisième poste conditionne les deux premiers, ou tout au moins, réduit les possibilités de compression à leur niveau. En effet si l'appel de chaînes de remplissage de pages squelettes ou de validation du message partiel, ou encore l'usage de pages dynamiques, étaient tout à fait exceptionnels, on pourrait imaginer de réduire le coût des points 1 et 2 en "décentralisant l'arbre de capture", c'est-à-dire en portant la librairie des pages et la chaîne EXAC dans une ou plusieurs machines "frontales" plus petites qui délivreraient à la machine centrale des messages de transaction complets formés par descente dans l'arbre décentralisé. Les arbres développés jusqu'ici pour les applications Diogène sont caractérisés par l'appel de plusieurs chaînes d'application (remplissage, auxiliaires, validation MFL) au cours de toute descente. Si ces chaînes doivent être exécutées dans la machine centrale, la lourdeur du mécanisme de leur mise en exécution à partir de l'arbre décentralisé, semble exclure cette approche; sous réserve toutefois des perspectives offertes par les terminaux programmables; l'équivalent de certaines chaînes de validation MFL simples, par exemple, pourrait être joint aux informations de contrôle d'une page type-C, sous forme d'instructions exécutables par le terminal.

Il en résulterait une réduction du nombre d'interactions arbre décentralisé/chaînes d'applications, d'où un renforcement de l'intérêt d'un arbre décentralisé.

Une autre manière de diminuer l'impact du point 2 tout en préservant la flexibilité là où elle est nécessaire, consiste à rendre EXAC capable de servir plusieurs terminaux en parallèle, d'un seul point de contrôle : pendant la descente les terminaux sont alors servis par une partition résidant constamment en mémoire centrale; la mise en exécution de chaînes d'application, en cours de descente, à d'autres points de contrôle est notablement plus simple dans cette solution que ce n'est le cas avec un arbre décentralisé.

En résumé, malgré les résultats substantiels des efforts faits pour réduire la portée des trois points mentionnés, la prise de message par arbre de capture demeure relativement coûteuse du point de vue des ressources en matériel. Toutefois cet aspect des coûts est largement compensé par les économies réalisées sur d'autre plans : sécurité de la prise de message, efforts d'analyse, notamment.

4.4.3. Rôle des diverses tâches d'EXAC. Schéma bloc

Ce paragraphe présente le rôle des tâches formant la chaîne d'EXAC et schématise le passage du contrôle de l'une à l'autre.

Nom de tâche	Rôle
TSK1001	Tâche recevant, à titre exclusif, le contrôle de la part du monde extérieur; selon la valeur COD11, ..., COD20 (paragraphe 4.2.1), se débranche sur la tâche voulue.
TSK1002/ TSK1007	Mise en route de la descente. Ces tâches sont appelées par TSK1001 pour COD12/COD11 respectivement.
TSK1005	Exécute la boucle centrale de la descente en modes normal et correction, c'est-à-dire, une suite de sélections sans incident sur des pages de type-S.
TSK1003	Traitement des pages de type-C : envoi de l'affichage et décodage du message entré au clavier (MFL) en présence de cet affichage.
TSK1011	Descentes fictive et automatique.
TSK1012	Présentation de pages "clef" pour enchaînement de transactions.
TSK1017	Rétrogradation (touche fonction =).
TSK1019	Envoi d'EXAC4/5 après activation de la touche fonction "*" en mode normal.
TSK1021	Envoi d'EXAC1/3/7, affichages requis par des conditions particulières en cours de descentes réelle ou fictive.
TSK1015	Envoi d'EXAC2 pour demande de visa.
TSK1006	Traitement d'erreur; envoi d'EXAC8.
TSK1010	Tâche de traçage appelée par touche "?" (en période de test, permet l'examen du CB à tout moment pendant une descente).

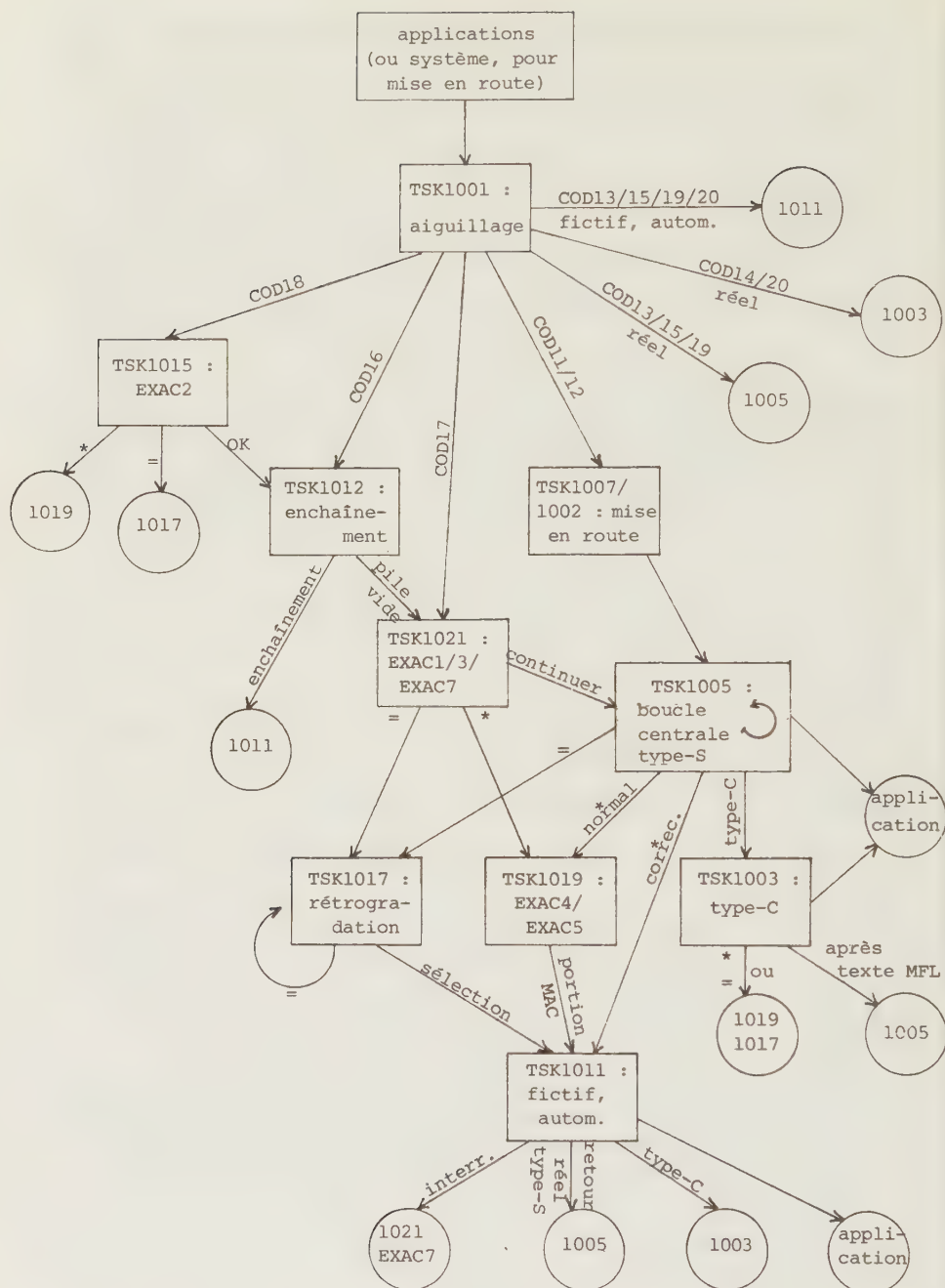


Fig. 4-3 Schéma bloc d'EXAC

Eclairons le schéma 4-3 de quelques commentaires.

Le premier message physique entré à un terminal inactif est transmis à EXAC par le système; on se trouve dans la situation décrite au paragraphe 4.3.3 de mise en route de la descente. La tâche d'aiguillage TSK1001 passe le contrôle à TSK1002 pour initialiser les piles, messages et zone sémaphores dans le bloc de communication, déterminer l'affichage de repos ou préparer un embryon de pile désignant la liste des patients applicable à l'unité de soins où se trouve le donneur d'ordre, avant passage à TSK1005 pour envoi de l'une ou l'autre de ces pages.

La tâche TSK1005 contient la boucle centrale de descente dans l'arbre de capture. Cette boucle peut être attaquée en divers points selon la tâche donnant le contrôle à TSK1005. On sort de cette tâche s'il est fait usage d'une touche fonction ("=" ou "*"), s'il apparaît une page type-C ou s'il est fait appel à une chaîne d'application (remplissage d'une page squelette, chaîne auxiliaire, chaîne de validation sémantique/exécution).

Suivant la valeur donnée COD11, ..., COD20 par la chaîne d'application, au retour dans EXAC, le contrôle revient, toujours via TSK1001 (qui d'ailleurs s'assure de la cohérence entre code reçu et type de chaîne d'application qui avait été appelée), à la boucle centrale de TSK1005 par exemple, ou, s'il s'agissait d'une chaîne de validation sémantique/exécution, à la tâche TSK1015 pour envoi d'EXAC2. Dans ce dernier cas, si l'opérateur choisit OK, après mise à jour physique de la banque de données résultant d'une requête émise dans TSK1015, le contrôle passe à TSK1012 pour enchaînement de transactions sur page clef; en admettant que la pile résiduelle ne soit pas vide, et que l'opérateur fasse, sur une page clef, un choix autre que "sortie", une descente fictive s'exécute dans TSK1011 jusqu'à la page clef d'où la descente réelle reprend dans la tâche TSK1005.

Il est évident que ce schéma ne rend compte que des principaux débranchements possibles.

Chaque tâche est matérialisée par un "overlay" primaire dont le programme principal a pour nom celui de la tâche, et ne contient en général que relativement peu de code; celui-ci se trouve surtout dans des sous-programmes servant d'ordinaire dans plusieurs tâches.

4.5. Codification d'EXAC

4.5.1. Choix d'un langage

La chaîne EXAC est tout entière codée en COMPASS, l'assembleur des machines CDC 6000/CYBER. Deux raisons justifient ce choix :

- l'assembleur offre le plus de flexibilité en fait de paramétrisation, de génération de tables, d'interface avec le système, etc.;
- il donne les meilleures garanties d'efficacité en fait de volume occupé par le code, et de temps d'exécution.

Ces arguments, valables si l'on dispose d'un assembleur plutôt raffiné comme COMPASS [CDC 75], ont eu le pas sur ceux qu'on peut avancer en faveur de langages plus évolués :

1. portabilité : l'objectif à réaliser est un arbre de capture efficace dans l'environnement logiciel et matériel de Diogène; il paraît impossible de préserver l'efficacité sans tenir compte, par exemple, dans la conception même des tables, de l'existence de mots de 60 bits;
2. sécurité dans la programmation et aisance de mise au point et de stabilisation des programmes : selon notre expérience, l'essentiel des erreurs faites en codant en assembleur, du moins dans ce genre de programmes et sur la machine utilisée, proviennent de fautes dans les instructions de décalage et de masques sur des registres lorsqu'on veut extraire ou insérer un champ dans une table ou un registre. Or toutes ces erreurs peuvent être évitées en développant quelques macros qui permettent de définir les "coordonnées" d'un champ, puis de se référer à ce champ pour l'extraire ou l'insérer, en ne donnant que le nom qu'il a reçu lors de sa définition; du même coup, si l'on change la structure d'une table, il suffit en général de modifier la définition des champs et de recompiler;
3. programmation plus modulaire : la programmation en assembleur n'interdit pas la modularité et n'oblige pas à faire des branchements tous azimuts; d'ailleurs quelques branchements commentés nuisent moins à la clarté du code que cinq niveaux de "if... then... else" imbriqués.

4.5.2. Effort requis

L'ensemble de ce qui est décrit dans ces chapitres (et nombre d'éléments secondaires passés sous silence pour ne pas surcharger), a été développé sur trois ans et représente un effort de 6-7 hommes-années. Dans ce total nous avons déjà inclus

le développement de DESPAC et CHAPAC soit environ 2-2 1/2 hommes-années. Le reste peut se décomposer comme suit :

4 mois : production de spécifications externes pour DESPAC/CHAPAC/EXAC; premier "document de base", préalable à tout développement de code et même d'analyse interne, si ce n'est très générale;

2 mois : analyse interne générale d'EXAC;

3 mois : préparation de l'infrastructure : "texte", au sens COMPASS, centralisant la définition du contenu des tables formant les pages, définition utilisée à la fois par DESPAC, CHAPAC et EXAC; structure du CB (communication block) vu par EXAC; préparation des macros permettant d'exploiter ces structures; mise au point d'outils de traçage pour EXAC.

3-3 1/2 hommes-années :

développement d'EXAC, y-compris les perfectionnements apportés à l'infrastructure, les adaptations partielles de l'analyse, les améliorations demandées en cours de développement, la production d'une seconde version du "document de base", l'élimination des erreurs mises en évidence par l'utilisation que font du produit ceux qui développent les chaînes d'application.

A ce stade, la chaîne EXAC qui comporte environ 15'000 instructions actives, commence à présenter une stabilité et une qualité (fréquence de détection d'erreurs par des utilisateurs) satisfaisantes.

5. Bilan et perspectives

Il est peut-être prématuré de dresser un bilan de l'arbre de capture dans le projet Diogène : les applications constituant le SIH ne sont pas toutes écrites, ni, a fortiori, en production. Le seraient-elles que l'évaluation des avantages et des inconvénients du procédé de prise des messages se ferait par comparaison avec un projet Diogène sans l'arbre de capture, projet fictif évidemment. Par rapport à la conception de départ de l'arbre de capture pour Diogène, on constate que l'évolution générale du produit tend à en faire une interface universelle de saisie des messages, tendance que manifestent d'une part le renforcement du rôle des pages "squelettes" pour l'affichage d'informations variables extraites de la banque de données, d'autre part les développements très substantiels concernant les pages type-C (préremplissage, interactions améliorées avec le donneur d'ordre). Cette évolution est la conséquence d'un double phénomène :

- a) La première application mise en chantier a été l'enregistrement des patients. Or il s'agit d'une application visiblement peu faite pour la prise d'information par sélections sur des pages type-S fixes : il faut introduire un volume important de données inédites lors de l'admission de nouveaux patients, et dans tous les cas, des listes de patients déjà connus du SIH et les données qui les concernent, doivent apparaître à divers stades de l'enregistrement.
- b) L'arbre de capture a été pris au piège de son propre succès ; l'intérêt évident de concentrer en un seul produit tout ce qui concerne les interactions avec les terminaux, a conduit à étendre les possibilités de l'arbre de capture plutôt que d'y renoncer dans les cas où le processus de sélection est inadéquat.

Ainsi, sans remettre en cause aucune option fondamentale, la conception de l'arbre de capture a évolué parallèlement à une perception moins schématique des impératifs de la saisie des messages dans l'ensemble du projet Diogène. Cette faculté d'adaptation, jointe aux avantages de l'arbre de capture pour la prise de messages, avantages abondamment discutés aux chapitres précédents, permet de considérer l'arbre de capture comme un élément positif du projet Diogène.

Quant aux perspectives, diverses voies s'offrent aux développements futurs. De manière un peu arbitraire, on peut ici distinguer

- a) le plan pratique (ou court terme) : améliorations des performances (par exemple EXAC résident, servant plusieurs terminaux en parallèle) et extensions des possibilités des produits supportant l'arbre de capture;

b) le plan théorique (ou long terme) :

- prolongements des développements du chapitre 2 et, s'il y a lieu, mise à profit, dans l'arbre de capture Diogène, des constatations faites;
- investigations dans des domaines inexplorés, notamment l'usage de l'arbre de capture comme instrument de validation automatique de messages donnés a priori : il s'agit de rechercher une descente dans l'arbre de capture, (en admettant qu'une telle descente existe) générant le message donné ou un équivalent (à définir) de ce message; ceci permettrait par exemple, à des opérateurs très agiles, de court-circuiter le processus de descente par un message présenté en format libre et de valider ce message après coup. Il est à peine besoin de souligner l'ampleur et l'intérêt des problèmes que soulève cette étude.

R E F E R E N C E S

=====

Barnett et al. 68

G. Octo Barnett, R.A. Greenes : Interface aspects of a Hospital Information System (Annals of the New York Academy of Sciences, vol. 161, pp. 756-768, 1968)

Berney/Baud/Scherrer 77

J.P. Berney, R. Baud, J.R. Scherrer : Implementation of a frame selection system (Medinfo 77, North-Holland, 1977)

Berge 70

C. Berge : Graphes et hypergraphes (Dunod, 1970)

Bowden et al. 73

K.F. Bowden, G.I.L. Grafton, I.R. Mac Callum : A Method of sequential selection for information capture in a real-time information system (A. Günther et al. (eds), International Computing Symposium 1973, North-Holland, 1974)

CDC 70

Setran (Selectable Element Translator, MEDISCOPE)
Reference Manual (Control Data Publ. 60259400, 1970)

CDC 75

COMPASS Reference Manual (Control Data Publ. 60492600)

CDC 76

NOS (Network Operating System) Reference Manual, vol. 1
(Control Data Publ. 60435400)

Gilbert/Moore 59

E.N. Gilbert, E.F. Moore : Variable Length Binary Encoding
(Bell System Tech. J. 38, pp. 933-968, 1959)

Gross/Lentin 67

M. Gross et A. Lentin : Notions sur les grammaires formelles,
2ème édition (Gauthier-Villars, 1970)

Hughes 72

Michel Hughes : Initiation mathématique aux grammaires formelles
(Larousse, 1972)

Huffman 52

D.A. Huffman : A method for the construction of minimum redundancy codes (Proc. IRE, 40, pp. 1098-1101, 1952)

Khinchin 57

A.I. Khinchin : Mathematical Foundations of Information Theory
(Dover, 1957)

Knuth 68

Donald E. Knuth : The Art of Computer Programming, vol. 1 :
Fundamental Algorithms (Addison-Wesley, 1968)

Knuth 73

Vol. 3 : *Sorting and Searching* (Addison-Wesley, 1973)

Le Beux et al. 74

P.J. Le Beux, R.R. Henley, M.S. Blois : *Implementation of a Frame Selection System for a Modular Hospital Information System*.
Medinfo 74, pp. 625-631 (North-Holland, 1974)

Le Beux 75

P. Le Beux : *Les langages de grilles de sélection appliqués au langage médical* (Colloque IRIA 1975, VII, pp. 1-12)

Martinet 70

André Martinet : *Eléments de linguistique générale*
(Armand Colin, 1970)

Moreau 75

René Moreau : *Théorie des langages* (Hachette, 1975)

Nijssen 74

G.M. Nijssen : *Present and Future Possibilities of Data Base Technology*. Medinfo 74, pp. 1-4 (North-Holland, 1974)

Picard 72

C.F. Picard : *Graphes et questionnaires*,
Tome 2 : *Questionnaires* (Chap. 8-14) (Gauthier-Villars, 1972)

Prieto 75

Luis J. Prieto : *Pertinence et pratique. Essai de sémiologie*
(Les Editions de Minuit, 1975)

Scherrer/Baud 73

J.R. Scherrer, R. Baud : *Quelques-unes des contraintes caractéristiques et dominantes d'un système d'information hospitalière (SIH)* (Médecine et Hygiène, 31, pp. 1170-1173, 1973)

Schultz et al. 71

J.R. Schultz, S.V. Cantrill, K.G. Morgan : *An initial operational problem oriented medical record system-For storage, manipulation and retrieval of medical data* (AFIPS, Conference Proceedings, vol. 38, New Jersey, pp. 239-264, 1971)

Shannon 48

C.E. Shannon : *A mathematical theory of communication* (Bell System tech. J. 27, pp. 379-423, 623-656, 1948)

Shannon 65

C.E. Shannon : *Information Theory*, in *Encyclopaedia Britannica*, vol. 12, édition 1965

Publications internes de la Division d'informatique de l'Hôpital

Berney 75

Prise de messages par arbre de capture. Document de base,
seconde version (Cahier Diogène No 4)

Gilliard 76

ACAP - EXSACAP (Cahier Diogène No 7)

Mazzario 76

PREFTN - Manuel de référence (Cahier Diogène No 6)

Annexe A

Figures de la démonstration sur papier (paragraphe 1.2)

N.B. : Les affichages apparaissant dans cette démonstration ne sauraient être considérés comme un aperçu fidèle des applications traitées par le SIH Diogène. Il s'agit d'ébauches illustrant certains concepts.

EXEMPLE 1

```

S      *****
      5CL
      ** PATIENTS DE L'UNITE **
      *
      *B AEBISCHER ARMAND      M      37      N
      *
      *C ASSIMACOPOULOS ALAIN      M      36      O
      *
      *D BANDERET ANDRE      M      36      P
      *
      *E BARBE ALBERT      M      34      Q
      *
      *F BAUD ALFRED      M      33      R
      *
      *G BERNEY ANDREE      F      32      S
      *
      *H CLERC ANNE-MARIE      F      31      T
      *
      *I DUPONT CLAUDE      M      37      U
      *
      *J      V
      *
      *      *****
      *      M ABANDONNER
      *
      *      X FONCTIONS POUR D'AUTRES PATIENTS
      *
      *****

```

On suppose être dans l'unité 5CL. Le premier affichage est la liste des patients de l'unité. Hors du cadre, un "S" signale qu'il s'agit d'une page de sélection (type-S), où l'opérateur peut faire le choix d'une information. En haut à gauche de l'écran, "N" signale une descente normale dans l'arbre.

L'opérateur a choisi la position marquée par >>>>

PAT+1

S

* NK
#5CL ,DUPONT CLAUDE M 37 ,PRESCR. DE SURVEILLANCE PARTICULIER
#E

*>>>>C CREATION
#D ANNULATION

X

Sur cet affichage le choix étiqueté "X" est masqué : blanchi et insensible. Son contenu sera visible dans un affichage ultérieur de cette page. L'idée est ici d'obliger l'opérateur à choisir une opération plutôt que la sortie lors de la première apparition de la page.

GES03+1

[illegible]

5

GES03*2

```

* * * * *
* N
* SCL , DUPONT CLAUDE
* E , CONTROLE AU LEVER
*
* VOULEZ-VOUS CONTROLER AU LEVER
* C SON ASPECT
*
* >>>>> D SON EQUILIBRE
*
* E SA CAPACITE DE SE MOUVIR
*
* * * * *

```

M 37 , PRESCR. DE SURVEILLANCE PARTICULIER

5

○
+
○
○
○
○

N M 37 ,PRESCR. DE SURVEILLANCE PARTICULIER
#SCL ,DUPONT CLAUDE
#E,CONTROLE AU LEVER,SON EQUILIBRE

4C 1 FOIS PAR JOUR

#D DEUX FOIS PAR JOUR

*E TROIS FCIS PAR JOUR

#F UNE FCIS PAR SEMAINE

*6*DEUX FOIS PAR SEMAINE

#H TROIS FOIS PAR SEMAINE

QUAND C'EST NECESSAIRE

X CONTINUUELLEMENT

S

TEMP S+1

* N

*5CL ,DUFONT CLAUDE

*E,CONTROLE AU LEVER,SON EQUILIBRE,DEUX FOIS PAR JOUR

M 37

,PRESCR. DE SURVEILLANCE PARTICULIER

>>>>>*C DES AUJOURD'HUI

*D DES DEMAIN

*E AUTRE

Le donneur d'ordre choisit de produire une autre prescription de surveillance particulière pour Dupont Claude. S'il avait choisi "X" sur la page clef, la page clef de niveau supérieur serait apparue, comme on peut le voir plus loin.

S

```

*****
* N
* 5CL ,DUPONT CLAUDE
* 5E ,ALIMENTATION
*
* 5C CONTROLER L'APPETIT
*
* 5D COMPORTEMENT AU REPAS
*
>>>> 5E SONDE DE GAVAGE
*
* 5F ENTRE LES REPAS
*
* 5G CONTROLE DES POISSONS
*****

```

GE503+4

5

TEMPS+1

* N
 * 5CL • DUPONT CLAUDE
 M 37 , PRESQ. DE SURVEILLANCE PARTICULIER

M 37 ,PRESCR. DE SURVEILLANCE PARTICULIER
SAVAGE,CONTINUUELLEMENT

*C DES AUJOURD'HUI

*D DES DEMAIN

>>>>E AUTRE

```
*****
* N 5CL ,DUPONT CLAUDE N 37 ,PRESCR. DE SURVEILLANCE P
* ARTICULIERE,ALIMENTATION,SONDE DE GAVAGE,CONTINUUELLEMENT
*
* QUEL JOUR ?
* <JOUR> [REDACTED]
* <MOIS> [REDACTED]
* <ANNEE> [REDACTED]
*
* <>N*INSCRIRE QUE DES CHIFFRES
*
*****
```

Les dates courantes "aujourd'hui" ou "demain" ne lui convenant pas, le donneur d'ordre entre une date à la carte sur une page cliqué ou de type-C (remarquer le C hors du cadre, à gauche), par opposition aux pages ordinaires, de type-S, permettant les sélections. L'opérateur entre les informations dans des champs spécifiés comme non protégés par l'analyste rédacteur de pages (champs hachurés ci-dessus); ces champs peuvent être présentés par exemple en "video inverse". Les parenthèses angulaires n'apparaissent pas sur l'écran.


```

** R
** #5CL , DUPONT CLAUDE M 37 , PRESCK. DE SURVEILLANCE PARTICULIER
** #E, ALIMENTATION, SONDE DE GAVAGE, CONTINUUELLEMENT, JOUR = 23, MOIS = 12, ANNEE = 76
**
** ---- DUREE ---
** #C UN JOUR
**
** #D DEUX JOURS
**
** #E TROIS JOURS
**
** #F UNE SEMAINE
**
** #G PENDANT TOUT LE SEJOUR

```

"R" en haut à gauche de l'écran indique la rétrogradation. On pourrait continuer à reculer, toujours avec la touche "=", mais le donneur d'ordre décide de ne changer que la durée. Il reprend la descente en faisant une sélection sur la page présentée en rétrogradation. Notez que le MAC se rétrécit en cours de recul.

C

TEMPS*3

```

*****
* N
* SCL ,DUPLNT CLAUDE
* E,ALIMENTATION,SONDE DE GAVAGE,CONTINUUELLEMENT,JOUR=23,MOIS=12,ANNEE=76
*
* DUREE DE LA SURVEILLANCE
*
* <NOMBRE DE JOURS>
*****

```

Page type-C permettant d'introduire une durée de surveillance. Le libellé "durée de surveillance" peut être spécifié lors de l'appel de cette page, ce qui permet son utilisation dans divers contextes sans perdre de spécificité. Ceci vaut pour toute page de type-S ou -C.

GF503+0

```

S
*****
# NK
#5CL ,DUPONT CLAUDE      M 37 ,PRESCR. DE SURVEILLANCE PARTICULIER
#E
#C CREATION
#D ANNULLATION
*****
X PAS D'AUTRE TRANSACTION
*****
<<<<

```

Trois pages clef vont défiler avant que l'opérateur décide de faire autre chose que de sortir de ces pages.

5

FONCTI+0

5CL, DUPONT CLAUDE M 37

43 JV 4 EVT

GESTION DE SEJOUR

#F PREAVIS DE SORTIE
#(PAT. PRESENT VA SORTIR)

** SORTIE EFFECTIVE
 ** (PAT. QUITTE L'UNITE, FICHE D'ACCOMPAGNEMENT)

X RIEN D'AUTRE

五

5

2

**B1. TRANSCENNE JEAN

44

4B ADMISSION

48

* C. LABORATIF E

45

RADIOLOGIE

45

>>>>E PHARMACIE

48

•

4

4

✻

✻

48

• •

✱

+

22

51

•

4

5

11, TRANSCENNE JEAN

>>>>C PRESCRIPTION LISTE

**E PRESCRIPTION HORS LISTE

LOG CONSULTATION DU DICTIONNAIRE

PHAC

* * * * *
* N P1,TRANSCENNE JEAN,PHARMACIE LISTE
* * * * ** * * * *
* --- STUPEFIANTS ---
* * * * *>>>>>> * D OPIUM
* * * * ** E MORPHINE
* * * * ** F CODEINE
* * * * ** G PETHIDINE DCI
* * * * *

P MORPHINE-ATROPINE

L FENTANYL DCI

R THALAMONAL *

S

* N BI,TRANSCENNE JEAN,PHARMACIE LISTE,OPIUM *

* --- OPIUM --- *
* IN: COMME REGULATEUR DE LA MOTRICITE INTESTINALE. *
* LES PROPRIETES DE REGULATION DE LA MOTRICITE INTESTINALE *
* SONT UTILISEES DANS LA LUTTE CONTRE LES DIARRHEES *
* PR: SOUS FERME DE TINCTURA OPII CRUCATA = LAUCANUM *
* SOLUTION ALCOOLIQUE A 10% D'OPIUM = 1% DE MORPHINE *
* 35 GTT = 1 ML = 10 MG DE MORPHINE *
* DM: DMS 2.0 ML DMD 6.0 ML PC: 0.6 ML - 1.5 ML *
* DOSE PAR PRISE: *
>>>>SF 0.6 ML = 21 GTT = 6 MG K 1.2 ML = 42 GTT = 12 MG *
* G 0.8 ML = 28 GTT = 8 MG S 1.5 ML = 52 GTT = 15 MG *
* H 1.0 ML = 35 GTT = 10 MG T AUTRE (EN MG) *

PHA7

C

PHAI C

F1, TRANSCENNE JEAN, PHARMACIE LISTE, OPIUM, 0.6 ML = 21 GTT = 6 MG

TAPEZ LE CHIFFRE CORRESPONDANT S.V.P.

#<X FCIS PAR JOUR> 

* N P1,TRANSCENNE JEAN,PHARMACIE LISTE,OPIMUM, 0.6 ML = 21 GTT = 6 MG,12
* FOIS PAR JOUR

* --- EN RELATION AVEC QUEL GENRE DE PROBLEME ? ---

>>>>>D SUSPICION D'INFARCTUS

*E BILAN MEDICAL

EX 4

... Sur ce MAC éclaté, l'opérateur choisit la portion de message à corriger; EXAC envoie l'affichage type-S ou -C sur lequel la portion visée du message avait été produite au cours de la descente antérieure. De là, la descente, après que l'opérateur a introduit la correction voulue, peut se poursuivre soit "à la main", soit de manière automatique lorsque le chemin corrigé rejoint le chemin suivi avant correction.

10
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100

1.1, TRANSCENNE JEAN, PHARMACIE LISTE, 0.6 ML = 21 GTT = 6 NG

TAPEZ LE CHIFFRE CORRESPONDANT S.V.P.

<X FUIS FAF, JEUK>

La descente reprend donc au point visé sur EXAC4, en mode correction, signalé par un "C" en haut à gauche de l'écran. Il s'agit ici d'une page de type-C sur laquelle l'opérateur introduit la valeur corrigée "3".

S

PROB+3

C R1,TRANSCENNE JEAN,PHARMACIE LISTE,OPIUM, 0.6 ML = 21 GTT = 6 MG,3
FOIS PAR JOUR

--- EN RELATION AVEC QUEL GENRE DE PROBLEME ? ---

>>>>>D SUSPICION D'INFARCTUS

*E BILAN MEDICAL

Annexe B

Terminologie des graphes

La définition des termes utilisés en théorie des graphes ne fait pas l'unanimité des auteurs et des traducteurs; de là cette liste de définitions inspirées de [Berge 70] et, pour les deux dernières, de [Picard 72].

Graphe, sommet, arc, extrémités initiale et terminale : De manière très intuitive, un graphe est constitué par un ensemble de points (ou sommets) et un ensemble de flèches (ou arcs), reliant chacune deux sommets, ses extrémités initiale et terminale. Notation d'un graphe : (X, Γ) , où X désigne l'ensemble des sommets, et Γ celui des arcs.

Sommets adjacents : Deux sommets sont adjacents s'ils sont les extrémités d'un même arc.

Chaîne : Séquence d'arcs telle que chaque arc de la séquence ait une extrémité en commun avec l'arc précédent et l'autre extrémité en commun avec l'arc suivant.

Chemin : Chaîne particulière où tout arc a pour extrémité initiale l'extrémité terminale de l'arc précédent, et pour extrémité terminale l'extrémité initiale de l'arc suivant (chaîne dont tous les arcs se parcourent dans le même sens).

Cycle : Chaîne dont les deux extrémités coïncident, et où le même arc ne figure qu'une fois (chaîne fermée).

Circuit : Chemin fermé.

Graphe connexe : Graphe dont toute paire de sommets forme les extrémités d'une chaîne au moins.

Arbre : Graphe connexe sans cycle.

Racine : Sommet "a" d'un graphe, tel que tout autre sommet du graphe puisse être atteint par un chemin issu de "a".

Arborescence : Arbre n'admettant qu'une racine.

Graphe biparti : Graphe dont l'ensemble des sommets est partitionné en deux classes telles que deux sommets de la même classe ne soient jamais adjacents.

Valuation d'un graphe : Application de l'ensemble des arcs d'un graphe dans un ensemble de "valeurs", par exemple les réels.

Graphe valué : Triplet noté (X, Γ, V) où V est une valuation de l'ensemble Γ des arcs du graphe noté (X, Γ) .

Annexe C

Répertoire de sigles; glossaire, index

Terme	Signification	Références
AC	<u>Arbre de Capture</u> : au sens strict, librairie de pages; plus généralement, cette librairie et la chaîne (EXAC) qui l'exploite; au sens large, méthode de prise de messages par sélection d'information sur une suite d'affichages	1.2, annexe A
ACAP	<u>Arbre de Capture/APplication</u> : nom de la syntaxe à laquelle se conforme le MBG, message généré par descente dans l'AC au profit des applications	1.4.7, 2.3.1
CB	<u>Communication Block</u> : zone de mémoire centrale permettant aux tâches de communiquer entre elles	1.4.3, 1.4.7, 4.3
CHAPAC	<u>CHARGEur de Pages de l'AC</u> : utilitaire de création, d'entretien et d'analyse de librairies de pages préalablement compilées par DESPAC	1.4.5, 3.5
clef	Page clef : page réaffichée après exécution d'une transaction, par exemple pour faciliter la production d'un message variante du précédent (enchaînement de transactions)	annexe A, 1.4.4
correction de message	Modification apportée au message de transaction soit après rétrogradation soit après affichage du message éclaté	1.4.8, 4.3.7, 4.3.8
DESPAC	<u>DEScriptio</u> n de <u>Pages</u> de l' <u>AC</u> : format source des pages de l'AC; désigne aussi le programme transformant ce format source en tables compactes propres à la formation de librairies de pages	1.4.5, 3.3

Diogène	Division d'Informatique de l'HOpital cantonal de <u>GENE</u> ve : nom d'un SIH	introduction
dynamique	page dynamique : page dont le contenu, dans la librairie de pages, évolue au cours de la production en temps réel	1.4.4
enchaînement de transactions	voir "clef"	4.3.6
EXAC	<u>EX</u> ploitation de l' <u>AC</u> : chaîne de tâches permettant la descente dans l' <u>AC</u>	1.4.6, chap. 4
EXACi (i=1,...8)	affichages spéciaux, partie intégrante d'EXAC, présentés pour la demande de visa, la correction de message ou l'annonce d'erreur	4.1.3, annexe A
fictive	descente fictive : descente dans l' <u>AC</u> effectuée sans intervention de l'opérateur, permettant de reprandre la descente réelle sur une page rencontrée antérieure- ment	1.4.8
info-BG	<u>in</u> formation <u>B</u> ack <u>G</u> round : con- tribution apportée au MBG par une sélection	1.4.1
KWIC	<u>Key-Word-In-Context</u> : index de mots clef présentés dans leur contexte	2.2.2.3
MAC	<u>M</u> essage- <u>A</u> rbre de <u>C</u> apture : forme du message de transaction présen- tée à l'opérateur du terminal	1.4.1
MBG	<u>M</u> essage- <u>B</u> ack <u>G</u> round : forme du message de transaction transmise à la chaîne de tâches (program- mes) d'application exécutant la transaction (mise à jour de la banque de données)	1.4.1
message de transaction	information constituant la donnée nécessaire à l'exécution d'une transaction	1.1, 1.4.1
MFL	<u>M</u> essage en <u>F</u> ormat <u>L</u> ibre : infor- mation introduite par l'opéra- teur sur une page type-C, incor- porée, moyennant quelques mani- pulations par EXAC ou MAC et/ou	annexe A, 1.4.3

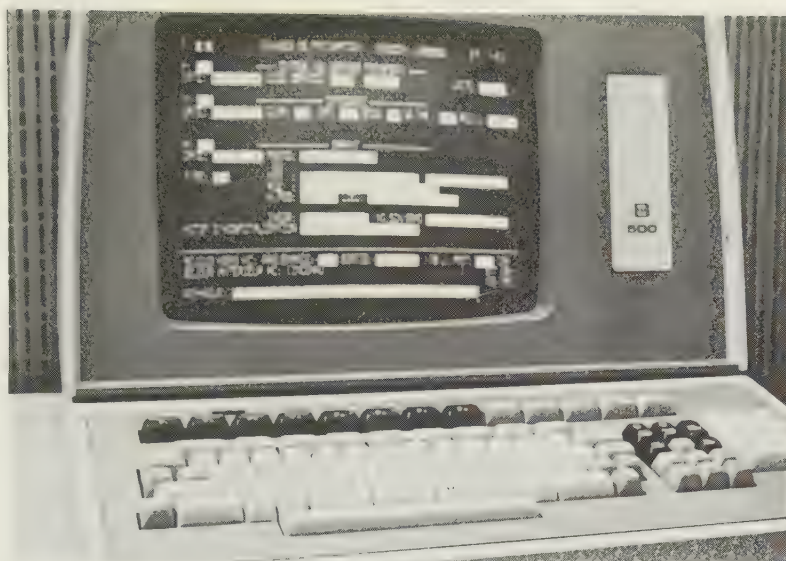
MMS	<u>Message-Myosotis</u> : forme du message de transaction à l'usage interne d'EXAC, trace de la descente effectuée dans l'AC	1.4.1
page	article du fichier librairie de pages contenant un affichage et des informations de contrôle, par exemple la désignation de pages successeurs pour les diverses sélections, les info-BG, etc.	1.4.1, chap. 3
registres software	variables permettant de désigner une page comme successeur à une grandeur (dont le contenu est déterminé au cours de la descente) additive près	1.4.5, 1.4.6
remplissage	chaîne de remplissage : chaîne d'application recherchant des informations dans la banque de données pour les présenter, via EXAC, dans une page squelette, type-S ou -C	1.4.3
SIH	Système d'Information Hospitalier : système transactionnel dévolu à la saisie, au traitement et à la diffusion de l'information dans un hôpital	1.1
squelette	page squelette : page dont le contenu provient partiellement d'informations prises de la banque de données par l'intermédiaire d'une chaîne de remplissage appelée avant envoi de l'affichage. Le contenu de la page en librairie n'évolue pas en temps réel, contrairement à celui des pages dynamiques	1.4.4
tâche	module entrant dans une "chaîne" exécutée en temps réel; a la forme d'un overlay (programme principal et routines) communiquant avec les autres tâches par la zone mémoire CB	1.4.3, 4.4
TL	Task-List : chaîne de tâches en général, chaîne d'application par opposition à EXAC	1.4.3

transaction	processus par lequel la banque de données d'un système transactionnel passe d'un état cohérent à un autre état cohérent	1.1
type-C	caractérise les pages <u>C</u> liché, sur l'affichage desquelles l'opérateur introduit des informations (MFL) dans des champs réservés à cet effet (non protégés)	annexe A, 1.3, 1.4.4
type-S	caractérise les pages de <u>S</u> élection, sur l'affichage desquelles l'opérateur peut faire des choix en cours de descente	annexe A, 1.3, 1.4.4, 3.3
visa	accord donné par le donneur d'ordre à l'exécution d'une transaction dont le message (MAC) apparaît sur un affichage (EXAC2) réservé à cet usage. Provoque la mise à jour physique de la banque de données	1.4.2, 4.1.4

Echantillon photographique

1. Terminal Beehive B500 affichant une page type-C
2. Affichage type-S
- 3-6 Deux affichages types-C

Chacun est reproduit deux fois pour faire ressortir successivement le contenu protégé et non protégé (vidéo inverse) de l'affichage.



- 1 -

DK FONCTIONS ADMISSION

SI ON CONNAIT LE NOM, PRENOM, DATE DE NAISSANCE, ET SEXE:

c ENREGISTREMENT	o RECHERCHE
d MISE A JOUR DU VSA	p CORRECTION DU VSA
e VISAS DE VSA	q CONSULTATION DE VSA
f MODIFICATION D'IDENTITE	

SINON:

h ENREGISTREMENT D'UN AMBULANT	t ENREGISTREMENT D'UN INCONNU
i ENREGISTREMENT SPECIAL PEDIATRIE	u ENREGISTREMENT PROVISOIRE
v TRAITEMENT DES DROITS D'ACCES	w VERIFICATION PL. ACCES
	x AUCUNE FONCTION

- 2 -

IDENTITE
LES RUBRIQUES 2 A 5 SONT OBLIGATOIRES

ENREGISTREMENT

PRENOM ☐ NOM ☐

PRENOM ACTUEL ☐ NOM ACTUEL ☐

PRENOM OFFICIEL ☐ NOM OFFICIEL ☐ (SEPARER PAR DES VIRGULES)

SEXE (M OU F OU I) ☐

DATE DE NAISSANCE: JOUR ☐ MOIS ☐ ANNEE ☐

PRENOM NON SE DIFFERENT DU PREMIER ☐

- 3 -

IDENTITE
LES RUBRIQUES 2 A 5 SONT OBLIGATOIRES

ENREGISTREMENT

PRENOM ☐ NOM ☐

PRENOM ACTUEL ☐ NOM ACTUEL ☐

PRENOM OFFICIEL ☐ NOM OFFICIEL ☐ (SEPARER PAR DES VIRGULES)

SEXE (M OU F OU I) ☐

DATE DE NAISSANCE: JOUR ☐ MOIS ☐ ANNEE ☐

PRENOM NON SE DIFFERENT DU PREMIER ☐

- 4 -

F: [REDACTED] DONNEES DE FACTURATION DOSSIER: (1000006) STA: (N-210)

H: [REDACTED] --- ENTREE (DU DERNIER TRANSFERT) ---

DATE: [REDACTED] CODE SERVICE: [REDACTED] [REDACTED] UNITE: [REDACTED]

H: [REDACTED] --- CATEGORIE ---

DATE: [REDACTED] CLAS: [REDACTED] CAT: [REDACTED] ORIG: [REDACTED] GR.TAR.: [REDACTED] PRIX/J: [REDACTED]

H: [REDACTED] --- GARANT ---

DATE: [REDACTED] MNEO: [REDACTED]

OU

TYPE: [REDACTED] NOM: [REDACTED]

RUE: [REDACTED]

N.POS: [REDACTED] LIEU: [REDACTED]

INCOR: [REDACTED] NO.SIN/GRAD: [REDACTED]

MOTIF D'HOSPITALISATION: [REDACTED] [REDACTED]

SEJOUR HORS HC: PROVENANCE: [REDACTED] SORTIE: [REDACTED] NS J. FACT: [REDACTED] VISA

SEJOUR ANTERIEUR HC: (INCONNU)) ENT (GEN)

REMARQUE: [REDACTED] MOO (MAR)

- 5 -

F: [REDACTED] DONNEES DE FACTURATION DOSSIER: (1000006) STA: (N-210)

H: [REDACTED] --- ENTREE (DU DERNIER TRANSFERT) ---

DATE: (13/06/77) CODE SERVICE: (LCH) (LCH) UNITE: (LCH)

H: [REDACTED] --- CATEGORIE ---

DATE: (13/06/77) CLAS: (6) CAT: (01) ORIG: (0) GR.TAR.: (01) PRIX/J: (210.00)

H: [REDACTED] --- GARANT ---

DATE: (13/06/77) MNEO: [REDACTED]

OU

TYPE: (5) NOM: (SAM) [REDACTED]

RUE: (24 RUE MICHEL DU CREST) [REDACTED]

N.POS: (1204) LIEU: (GENEVE) [REDACTED]

INCOR: [REDACTED] NO.SIN/GRAD: [REDACTED]

MOTIF D'HOSPITALISATION: (AFFECTION MEDICALE) [REDACTED]

SEJOUR HORS HC: PROVENANCE: [REDACTED] SORTIE: [REDACTED] NS J. FACT: [REDACTED] VISA

SEJOUR ANTERIEUR HC: (INCONNU)) ENT (GEN)

REMARQUE: (MAUVAIS PAYEUR) [REDACTED] MOO (MAR)

- 6 -

